Innovation, Sciences et Développement économique Canada
Office de la Propriété Intellectuelle du Canada

Innovation, Science and Economic Development Canada
Canadian Intellectual Property Office

CA 2916679 C 2021/06/15

(11)(21) 2 916 679

(12) BREVET CANADIEN
CANADIAN PATENT
(13) C

(86) Date de dépôt PCT/PCT Filing Date: 2014/07/16
(87) Date publication PCT/PCT Publication Date: 2015/01/22
(45) Date de délivrance/Issue Date: 2021/06/15
(85) Entrée phase nationale/National Entry: 2015/12/22
(86) N° demande PCT/PCT Application No.: US 2014/046875
(87) N° publication PCT/PCT Publication No.: 2015/009843
(30) Priorités/Priorities: 2013/07/18 (US61/847,951);
2014/07/15 (US14/332,181)

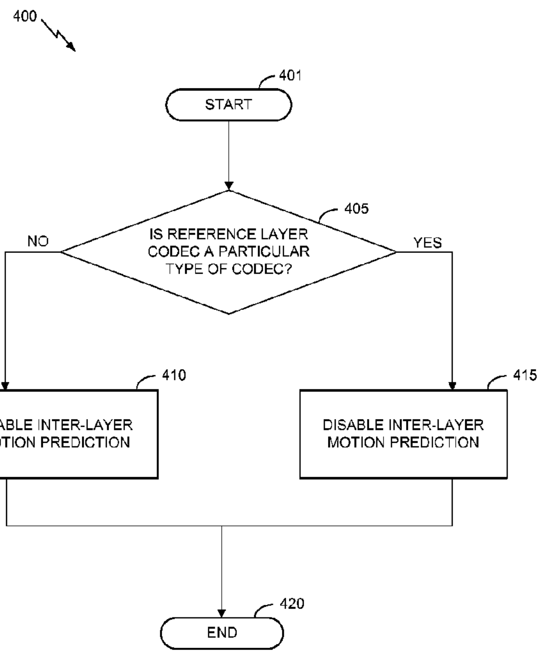
(51) Cl.Int./Int.Cl. *H04N 19/33* (2014.01), *H04N 19/51* (2014.01), *H04N 19/70* (2014.01), *H04N 19/187* (2014.01), *H04N 19/105* (2014.01)

(72) Inventeur/Inventor:
SEREGIN, VADIM, US

(73) Propriétaire/Owner:
QUALCOMM INCORPORATED, US

(74) Agent: SMART & BIGGAR LLP

(54) Titre : DISPOSITIF ET PROCEDE PERMETTANT LE CODAGE EXTENSIBLE D'INFORMATIONS VIDEO
(54) Title: DEVICE AND METHOD FOR SCALABLE CODING OF VIDEO INFORMATION



(57) Abrégé/Abstract:
An apparatus configured to code video information includes a memory and a processor in communication with the memory. The memory is configured to store video information associated with a reference layer and an enhancement layer, the reference layer associated with a reference layer (RL) codec and the enhancement layer associated an enhancement layer (EL) codec. The processor is configured to determine whether the RL codec associated with the reference layer is a particular type of codec, and in response to determining that the RL codec is a particular type of codec, process, in a video bitstream, an indication that motion information of the reference layer cannot be used to code the enhancement layer. The processor may encode or decode the video information.

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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
22 January 2015 (22.01.2015)

WIPO | PCT

(10) International Publication Number
WO 2015/009843 A1

(51) International Patent Classification:
H04N 19/33 (2014.01) *H04N 19/105* (2014.01)
H04N 19/51 (2014.01) *H04N 19/187* (2014.01)
H04N 19/70 (2014.01)

(21) International Application Number: PCT/US2014/046875

(22) International Filing Date: 16 July 2014 (16.07.2014)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/847,951 18 July 2013 (18.07.2013) US
14/332,181 15 July 2014 (15.07.2014) US

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

(72) Inventor: **SEREGIN, Vadim**; 5775 Morehouse Drive, San Diego, California 92121-1714 (US).

(74) Agent: **ABUMERI, Mark M.**; Knobbe Martens Olson & Bear LLP, 2040 Main Street, Fourteenth Floor, Irvine, California 92614 (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, 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, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, 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, 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**: DEVICE AND METHOD FOR SCALABLE CODING OF VIDEO INFORMATION

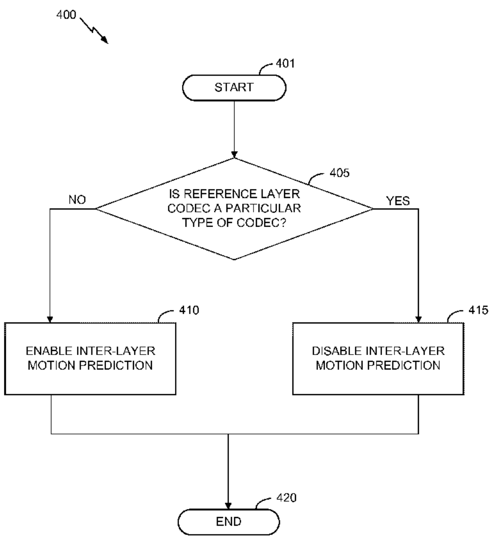


FIG. 4

(57) Abstract: An apparatus configured to code video information includes a memory and a processor in communication with the memory. The memory is configured to store video information associated with a reference layer and an enhancement layer, the reference layer associated with a reference layer (RL) codec and the enhancement layer associated an enhancement layer (EL) codec. The processor is configured to determine whether the RL codec associated with the reference layer is a particular type of codec, and in response to determining that the RL codec is a particular type of codec, process, in a video bitstream, an indication that motion information of the reference layer cannot be used to code the enhancement layer. The processor may encode or decode the video information.

WO 2015/009843 A1

DEVICE AND METHOD FOR SCALABLE CODING OF VIDEO INFORMATION

TECHNICAL FIELD

[0001] This disclosure relates to the field of video coding and compression, particularly to scalable video coding (SVC), multiview video coding (MVC), or 3D video coding (3DV).

BACKGROUND

[0002] 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, digital cameras, digital recording devices, digital media players, video gaming devices, video game consoles, cellular or satellite radio telephones, video teleconferencing devices, and the like. Digital video devices implement video compression 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) standard presently under development, 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.

[0003] Video compression techniques perform 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, a portion of a video frame, etc.) may be partitioned into video blocks, which may also be referred to as treeblocks, coding units (CUs) and/or coding nodes. Video blocks in an intra-coded (I) slice of a picture are encoded using spatial prediction with respect to reference samples in neighboring blocks in the same picture. Video blocks in an inter-coded (P or B) slice of a picture may use spatial prediction with respect to reference samples in neighboring blocks in the same picture or temporal prediction with respect to reference samples in other reference pictures. Pictures may be referred to as frames, and reference pictures may be referred to as reference frames.

[0004] 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. An inter-coded block is encoded according to a motion vector that points to a block of reference samples forming the predictive block, and the residual data indicating the difference between the coded block and the predictive block. An intra-coded block is encoded according to an intra-coding mode and the residual data. 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. The quantized transform coefficients, initially arranged in a two-dimensional array, may be scanned in order to produce a one-dimensional vector of transform coefficients, and entropy encoding may be applied to achieve even more compression.

SUMMARY

[0005] Scalable video coding (SVC) refers to video coding in which a base layer (BL), sometimes referred to as a reference layer (RL), and one or more scalable enhancement layers (ELs) are used. In SVC, the base layer can carry video data with a base level of quality. The one or more enhancement layers can carry additional video data to support, for example, higher spatial, temporal, and/or signal-to-noise (SNR) levels. Enhancement layers may be defined relative to a previously encoded layer. For example, a bottom layer may serve as a BL, while a top layer may serve as an EL. Middle layers may serve as either ELs or RLs, or both. For example, a layer in the middle may be an EL for the layers below it, such as the base layer or any intervening enhancement layers, and at the same time serve as a RL for one or more enhancement layers above it. Similarly, in the Multiview or 3D extension of the HEVC standard, there may be multiple views, and information of one view may be utilized to code (e.g., encode or decode) the information of another view (e.g., motion estimation, motion vector prediction and/or other redundancies).

[0006] In SVC, a current block in the enhancement layer may be coded (e.g., encoded or decoded) using the information derived from a reference layer. For example, a current block in the enhancement layer may be coded using the information (e.g., texture information or motion information) of a co-located block in the reference layer (the term “co-located” as used in the present disclosure may refer to a block in another layer that corresponds to the same image as the current block, e.g., the block that is currently being coded). In some implementations, whether a particular reference layer is used to code an enhancement layer may be signaled as a flag or syntax element. If the flag or syntax element indicates that the particular reference layer is used to code the enhancement layer, another flag or syntax element may further be signaled to indicate what kind of information in the particular reference picture is used to code the enhancement layer: texture (pixel) information, motion information, or both.

[0007] In certain cases, a portion of the information in the reference layer may not be available for use in coding the enhancement layer. For example, in some implementations, if the reference layer is coded using a non-HEVC codec, the motion information of the reference layer may not be available to an HEVC codec to code the enhancement layer. In such a case, the enhancement layer may still be coded using the texture information of the reference layer, but the motion information of the reference layer cannot be used to code the enhancement layer.

[0008] By exploiting this dependence of the availability of certain types of information in the reference layer on the type of codec used for coding the reference layer, some of the processing that is performed to determine what type of information is derived from the reference layer may be omitted (e.g., if the information is unavailable, there is no need to check whether that information is used for coding the enhancement layer), thus resulting in improved coding efficiency and/or reduced computational complexity.

[0009] The systems, methods and devices of this disclosure each have several innovative aspects, no single one of which is solely responsible for the desirable attributes disclosed herein.

[0010] In one aspect, an apparatus configured to code video information includes a memory and a processor in communication with the memory. The memory is configured to store video information associated with a reference layer and an enhancement layer, the reference layer associated with a reference layer (RL) codec and the enhancement layer associated an enhancement layer (EL) codec. The processor is configured to determine whether the RL codec associated with the reference layer is a particular type of codec, and in response to determining that the RL codec is a particular type of codec, process, in a video bitstream, an indication that motion information of the reference layer cannot be used to code the enhancement layer. The processor may encode or decode the video information.

[0011] In one aspect, a method of coding (e.g., encoding or decoding) video information comprises determining whether a reference layer (RL) codec associated with a reference layer is a particular type of codec; and in response to determining that the RL codec is a particular type of codec, processing, in a video bitstream, an indication that motion information of the reference layer cannot be used to code an enhancement layer associated with an enhancement layer (EL) codec.

[0012] In one aspect, a non-transitory computer readable medium comprises code that, when executed, causes an apparatus to perform a process. The process includes storing video information associated with a reference layer and an enhancement layer, the reference layer associated with a reference layer (RL) codec and the enhancement layer associated an enhancement layer (EL) codec; determining whether the RL codec associated with the reference layer is a particular type of codec; and in response to determining that the RL codec is a particular type of codec, processing, in a video bitstream, an indication that motion information of the reference layer cannot be used to code the enhancement layer.

[0013] In one aspect, a video coding device configured to code video information comprises means for storing video information associated with a reference layer and an enhancement layer, the reference layer associated with a reference layer (RL) codec and the enhancement layer associated an enhancement layer (EL) codec; means for determining whether the RL codec associated with the reference layer is a particular type of codec; and means for processing, in a video bitstream, an indication that motion information of the reference layer cannot be used to code the enhancement layer, in response to determining that the RL codec is a particular type of codec.

[0013a] According to one aspect of the present invention, there is provided an apparatus configured to code video information, the apparatus comprising: a memory configured to store video data associated with a reference layer, an enhancement layer, and one or more additional layers, the reference layer associated with a reference layer (RL) codec and the enhancement layer associated an enhancement layer (EL) codec, the reference layer comprising a first information portion and a second information portion, wherein the first information portion comprises texture information and the second information portion comprises motion information; and a processor in communication with the memory, the processor configured to: determine whether the RL codec associated with the reference layer is a first type of codec; based on a determination that the RL codec is the first type of codec, signal or receive, for each respective layer of the one or more additional layers, one or more flags or syntax elements indicative of whether information of the respective layer can be used to code the enhancement layer; based on the determination that the RL codec is the first type of codec, refrain from signaling or receiving one or more flags or syntax elements indicative of whether information of the reference layer can be used to code the enhancement layer and infer that a flag or syntax element indicative of whether the information of the reference layer can be used to code the enhancement layer has a first value without signaling or receiving the flag or syntax

Date Reçue/Date Received 2020-10-02

element; and based on the determination that the RL codec is the first type of codec, code the enhancement layer using only the first information portion and not using the second information portion to code the enhancement layer.

[0013b] According to one aspect of the present invention, there is provided a method of coding video information, the method comprising: determining whether a reference layer (RL) codec associated with a reference layer is a first type of codec, the reference layer comprising a first information portion and a second information portion, wherein the first information portion comprises texture information and the second information portion comprises motion information; based on a determination that the RL codec is the first type of codec, signaling or receiving, for each respective layer of one or more additional layers, one or more flags or syntax elements indicative of whether information of the respective layer can be used to code an enhancement layer; based on the determination that the RL codec is the first type of codec, refraining from signaling or receiving one or more flags or syntax elements indicative of whether information of the reference layer can be used to code the enhancement layer and inferring that a flag or syntax element indicative of whether the information of the reference layer can be used to code the enhancement layer has a first value without signaling or receiving the flag or syntax element; based on the determination that the RL codec is the first type of codec, code the enhancement layer using only the first information portion and not using the second information portion to code the enhancement layer.

[0013c] According to one aspect of the present invention, there is provided non-transitory physical computer storage having stored thereon computer executable code that, when executed, causes an apparatus to: store video data associated with a reference layer, an enhancement layer, and one or more additional layers, the reference layer associated with a reference layer (RL) codec and the enhancement layer associated an enhancement layer (EL) codec, the reference layer comprising a first information portion and a second information portion, wherein the first information portion comprises texture information and the second information portion comprises motion information; determine whether the RL codec associated with the reference layer is a first type of codec; based on a determination that the RL codec is the first type of codec, signal or receive, for each respective layer of the one or more additional layers, one or more flags or syntax elements

Date Reçue/Date Received 2020-10-02

indicative of whether information of the respective layer can be used to code the enhancement layer; based on the determination that the RL codec is the first type of codec, refrain from signaling or receiving one or more flags or syntax elements indicative of whether information of the reference layer can be used to code the enhancement layer and infer that a flag or syntax element indicative of whether the information of the reference layer can be used to code the enhancement layer has a first value without signaling or receiving the flag or syntax element; based on the determination that the RL codec is the first type of codec code the enhancement layer using only the first information portion and not using the second information portion to code the enhancement layer.

[0013d] According to one aspect of the present invention, there is provided a video coding device configured to code video information, the video coding device comprising: means for storing video data associated with a reference layer, an enhancement layer, and one or more additional layers, the reference layer associated with a reference layer (RL) codec and the enhancement layer associated an enhancement layer (EL) codec, the reference layer comprising a first information portion and a second information portion, wherein the first information portion comprises texture information and the second information portion comprises motion information; means for determining whether the RL codec associated with the reference layer is a first type of codec; means for signaling or receiving, for each respective layer of the one or more additional layers one or more flags or syntax elements indicative of whether information of the respective layer can be used to code the enhancement layer based on a determination that the RL codec is the first type of codec, wherein the means for signaling or receiving is configured to: based on the determination that the RL codec is the first type of codec, refrain from signaling or receiving one or more flags or syntax elements indicative of whether information of the reference layer can be used to code the enhancement layer and infer that a flag or syntax element indicative of whether the information of the reference layer can be used to code the enhancement layer has a first value without signaling or receiving the flag or syntax element; and based on the determination that the RL codec is the first type of codec code the enhancement layer using only the first information portion and not using the second information portion to code the enhancement layer.

Date Reçue/Date Received 2020-10-02

[0013e] According to one aspect of the present invention, there is provided an apparatus configured to code video information, the apparatus comprising: a memory configured to store video data associated with a reference layer and an enhancement layer, the reference layer associated with a reference layer (RL) codec and the enhancement layer associated an enhancement layer (EL) codec, the reference layer comprising a first information portion and a second information portion, wherein the first information portion comprises texture information and the second information portion comprises motion information; and a processor in communication with the memory, the processor configured to: determine whether the RL codec associated with the reference layer is a first type of codec; based on a determination that the RL codec is the first type of codec, refrain from signaling or receiving one or more flags or syntax elements indicative of whether information of the reference layer can be used to code the enhancement layer, based on the determination that the RL codec is the first type of codec, infer that a flag or syntax element indicative of whether the information of the reference layer can be used to code the enhancement layer has a first value without signaling or receiving the flag or syntax element, and based on the determination that the RL codec is the first type of codec, code the enhancement layer using only the first information portion and not using the second information portion to code the enhancement layer.

BRIEF DESCRIPTION OF DRAWINGS

[0014] **FIG. 1A** is a block diagram illustrating an example video encoding and decoding system that may utilize techniques in accordance with aspects described in this disclosure.

[0015] **FIG. 1B** is a block diagram illustrating another example video encoding and decoding system that may perform techniques in accordance with aspects described in this disclosure.

[0016] **FIG. 2A** is a block diagram illustrating an example of a video encoder that may implement techniques in accordance with aspects described in this disclosure.

[0017] **FIG. 2B** is a block diagram illustrating an example of a video encoder that may implement techniques in accordance with aspects described in this disclosure.

Date Reçue/Date Received 2020-10-02

[0018] **FIG. 3A** is a block diagram illustrating an example of a video decoder that may implement techniques in accordance with aspects described in this disclosure.

[0019] **FIG. 3B** is a block diagram illustrating an example of a video decoder that may implement techniques in accordance with aspects described in this disclosure.

[0020] **FIG. 4** illustrates a flow chart illustrating a method of coding video information, according to one embodiment of the present disclosure.

[0021] **FIG. 5** illustrates a flow chart illustrating a method of coding video information, according to another embodiment of the present disclosure.

Date Reçue/Date Received 2020-10-02

DETAILED DESCRIPTION

[0022] Certain embodiments described herein relate to inter-layer prediction for scalable video coding in the context of advanced video codecs, such as HEVC (High Efficiency Video Coding). More specifically, the present disclosure relates to systems and methods for improved performance of inter-layer prediction in scalable video coding (SVC) extension of HEVC.

[0023] In the description below, H.264/AVC techniques related to certain embodiments are described; the HEVC standard and related techniques are also discussed. While certain embodiments are described herein in the context of the HEVC and/or H.264 standards, one having ordinary skill in the art may appreciate that systems and methods disclosed herein may be applicable to any suitable video coding standard. For example, embodiments disclosed herein may be applicable to one or more of the following standards: 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 Multiview Video Coding (MVC) extensions.

[0024] HEVC generally follows the framework of previous video coding standards in many respects. The unit of prediction in HEVC is different from that in certain previous video coding standards (e.g., macroblock). In fact, the concept of macroblock does not exist in HEVC as understood in certain previous video coding standards. Macroblock is replaced by a hierarchical structure based on a quadtree scheme, which may provide high flexibility, among other possible benefits. For example, within the HEVC scheme, three types of blocks, Coding Unit (CU), Prediction Unit (PU), and Transform Unit (TU), are defined. CU may refer to the basic unit of region splitting. CU may be considered analogous to the concept of macroblock, but it does not restrict the maximum size and may allow recursive splitting into four equal size CUs to improve the content adaptivity. PU may be considered the basic unit of inter/intra prediction and it may contain multiple arbitrary shape partitions in a single PU to effectively code irregular image patterns. TU may be considered the basic unit of transform. It can be defined independently from the PU; however, its size may be limited to the CU to which the TU belongs. This separation of the block structure into three different concepts may allow each to be optimized according to its role, which may result in improved coding efficiency.

[0025] For purposes of illustration only, certain embodiments disclosed herein are described with examples including only two layers (e.g., a lower layer such as the base layer, and a higher layer such as the enhancement layer). It should be understood that such examples may be applicable to configurations including multiple base and/or enhancement layers. In addition, for ease of explanation, the following disclosure includes the terms "frames" or "blocks" with reference to certain embodiments. However, these terms are not meant to be limiting. For example, the techniques described below can be used with any suitable video units, such as blocks (e.g., CU, PU, TU, macroblocks, etc.), slices, frames, etc.

Video Coding Standards

[0026] A digital image, such as a video image, a TV image, a still image or an image generated by a video recorder or a computer, may consist of pixels or samples arranged in horizontal and vertical lines. The number of pixels in a single image is typically in the tens of thousands. Each pixel typically contains luminance and chrominance information. Without compression, the quantity of information to be conveyed from an image encoder to an image decoder is so enormous that it renders real-time image transmission impossible. To reduce the amount of information to be transmitted, a number of different compression methods, such as JPEG, MPEG and H.263 standards, have been developed.

[0027] Video coding standards include 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 Multiview Video Coding (MVC) extensions.

[0028] In addition, a new video coding standard, namely High Efficiency Video Coding (HEVC), is being 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). The full citation for the HEVC Draft 10 is document JCTVC-L1003, Bross et al., "High Efficiency Video Coding (HEVC) Text Specification Draft 10," Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 12th Meeting: Geneva, Switzerland, January 14, 2013 to January 23, 2013. The multiview extension to HEVC, namely MV-HEVC, and the scalable extension to HEVC, named SHVC, are also being developed by the JCT-3V (ITU-T/ISO/IEC Joint Collaborative Team on 3D Video Coding Extension Development) and JCT-VC, respectively.

[0029] Various aspects of the novel systems, apparatuses, and methods are described more fully hereinafter with reference to the accompanying drawings. This disclosure may, however, be embodied in many different forms and should not be construed as limited to any specific structure or function presented throughout this disclosure. Rather, these aspects are provided so that this disclosure will be thorough and complete, and will fully convey the scope of the disclosure to those skilled in the art. Based on the teachings herein one skilled in the art should appreciate that the scope of the disclosure is intended to cover any aspect of the novel systems, apparatuses, and methods disclosed herein, whether implemented independently of, or combined with, any other aspect of the present disclosure. For example, an apparatus may be implemented or a method may be practiced using any number of the aspects set forth herein. In addition, the scope of the present disclosure is intended to cover such an apparatus or method which is practiced using other structure, functionality, or structure and functionality in addition to or other than the various aspects of the present disclosure set forth herein. It should be understood that any aspect disclosed herein may be embodied by one or more elements of a claim.

[0030] Although particular aspects are described herein, many variations and permutations of these aspects fall within the scope of the disclosure. Although some benefits and advantages of the preferred aspects are mentioned, the scope of the disclosure is not intended to be limited to particular benefits, uses, or objectives. Rather, aspects of the disclosure are intended to be broadly applicable to different wireless technologies, system configurations, networks, and transmission protocols, some of which are illustrated by way of example in the figures and in the following description of the preferred aspects. The detailed description and drawings are merely illustrative of the disclosure rather than limiting, the scope of the disclosure being defined by the appended claims and equivalents thereof.

[0031] The attached drawings illustrate examples. Elements indicated by reference numbers in the attached drawings correspond to elements indicated by like reference numbers in the following description. In this disclosure, elements having names that start with ordinal words (e.g., "first," "second," "third," and so on) do not necessarily imply that the elements have a particular order. Rather, such ordinal words are merely used to refer to different elements of a same or similar type.

Video Coding System

[0032] **FIG. 1A** is a block diagram that illustrates an example video coding system 10 that may utilize techniques in accordance with aspects described in this disclosure. As used described herein, the term "video coder" refers generically to both video encoders and video decoders. In this disclosure, the terms "video coding" or "coding" may refer generically to video encoding and video decoding.

[0033] As shown in **FIG. 1A**, video coding system 10 includes a source module 12 that generates encoded video data to be decoded at a later time by a destination module 14. In the example of **FIG. 1A**, the source module 12 and destination module 14 are on separate devices – specifically, the source module 12 is part of a source device, and the destination module 14 is part of a destination device. It is noted, however, that the source and destination modules 12, 14 may be on or part of the same device, as shown in the example of **FIG. 1B**.

[0034] With reference once again, to **FIG. 1A**, the source module 12 and the destination module 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, so-called "smart" pads, televisions, cameras, display devices, digital media players, video gaming consoles, video streaming device, or the like. In some cases, the source module 12 and the destination module 14 may be equipped for wireless communication.

[0035] The destination module 14 may receive the encoded video data to be decoded via a link 16. The link 16 may comprise any type of medium or device capable of moving the encoded video data from the source module 12 to the destination module 14. In the example of **FIG. 1A**, the link 16 may comprise a communication medium to enable the source module 12 to transmit encoded video data directly to the destination module 14 in real-time. The encoded video data may be modulated according to a communication standard, such as a wireless communication protocol, and transmitted to the destination module 14. The communication medium may comprise any wireless or wired communication medium, such as a radio frequency (RF) spectrum or one or more physical transmission lines. The communication medium may form part of a packet-based network, such as a local area network, a wide-area network, or a global network such as the Internet. The communication medium may include routers, switches, base stations, or any other equipment that may be useful to facilitate communication from the source module 12 to the destination module 14.

[0036] Alternatively, encoded data may be output from an output interface 22 to an optional storage device 31. Similarly, encoded data may be accessed from the storage device 31 by an input interface 28. The storage device 31 may include any of a variety of distributed or locally accessed data storage media such as a hard drive, 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 31 may correspond to a file server or another intermediate storage device that may hold the encoded video generated by the source module 12. The destination module 14 may access stored video data from the storage device 31 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 module 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. The destination module 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 31 may be a streaming transmission, a download transmission, or a combination of both.

[0037] The techniques of this disclosure are not limited to wireless applications or settings. The techniques 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, streaming video transmissions, e.g., via the Internet (e.g., dynamic adaptive streaming over HTTP (DASH), etc.), encoding of digital video for storage on a data storage medium, decoding of digital video stored on a data storage medium, or other applications. In some examples, video coding 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.

[0038] In the example of **FIG. 1A**, the source module 12 includes a video source 18, video encoder 20 and an output interface 22. In some cases, the output interface 22 may include a modulator/demodulator (modem) and/or a transmitter. In the source module 12, the video source 18 may include a source such as a video capture device, e.g., a video camera, a video archive containing previously captured video, a video feed interface to receive video from a video content provider, and/or a computer graphics system for generating computer graphics data as the source video, or a combination of such sources. As one example, if the video source 18 is a video camera, the source module 12 and the destination module 14 may form so-called camera phones or video phones, as illustrated in the example of **FIG. 1B**. However, the techniques described in this disclosure may be applicable to video coding in general, and may be applied to wireless and/or wired applications.

[0039] The captured, pre-captured, or computer-generated video may be encoded by the video encoder 20. The encoded video data may be transmitted directly to the destination module 14 via the output interface 22 of the source module 12. The encoded video data may also (or alternatively) be stored onto the storage device 31 for later access by the destination module 14 or other devices, for decoding and/or playback.

[0040] In the example of **FIG. 1A**, the destination module 14 includes an input interface 28, a video decoder 30, and a display device 32. In some cases, the input interface 28 may include a receiver and/or a modem. The input interface 28 of the destination module 14 may receive the encoded video data over the link 16. The encoded video data communicated over the link 16, or provided on the storage device 31, may include a variety of syntax elements generated by the video encoder 20 for use by a video decoder, such as the video decoder 30, in decoding the video data. Such syntax elements may be included with the encoded video data transmitted on a communication medium, stored on a storage medium, or stored a file server.

[0041] The display device 32 may be integrated with, or external to, the destination module 14. In some examples, the destination module 14 may include an integrated display device and also be configured to interface with an external display device. In other examples, the destination module 14 may be a display device. In general, the display device 32 displays the decoded video data to a user, and may comprise any of a variety of display devices such as a liquid crystal display (LCD), a plasma display, an organic light emitting diode (OLED) display, or another type of display device.

[0042] In related aspects, **FIG. 1B** shows an example video encoding and decoding system 10′ wherein the source and destination modules 12, 14 are on or part of a device or user device 11. The device 11 may be a telephone handset, such as a "smart" phone or the like. The device 11 may include an optional controller/processor module 13 in operative communication with the source and destination modules 12, 14. The system 10′ of **FIG. 1B** may further include a video processing unit 21 between the video encoder 20 and the output interface 22. In some implementations, the video processing unit 21 is a separate unit, as illustrated in **FIG. 1B**; however, in other implementations, the video processing unit 21 can be implemented as a portion of the video encoder 20 and/or the processor/controller module 13. The system 10′ may also include an optional tracker 29, which can track an object of interest in a video sequence. The object or interest to be tracked may be segmented by a technique described in connection with one or more aspects of the present disclosure. In related aspects, the tracking may be performed by the display device 32, alone or in conjunction with the tracker 29. The system 10′ of **FIG. 1B**, and components thereof, are otherwise similar to the system 10 of **FIG. 1A**, and components thereof.

[0043] Video encoder 20 and video decoder 30 may operate according to a video compression standard, such as the High Efficiency Video Coding (HEVC) standard presently under development, and may conform to a HEVC Test Model (HM). Alternatively, video encoder 20 and video decoder 30 may operate according to other proprietary or industry standards, such as the ITU-T H.264 standard, alternatively referred to as MPEG-4, Part 10, Advanced Video Coding (AVC), or extensions of such standards. The techniques of this disclosure, however, are not limited to any particular coding standard. Other examples of video compression standards include MPEG-2 and ITU-T H.263.

[0044] Although not shown in the examples of **FIGS. 1A** and **1B**, video encoder 20 and video decoder 30 may each be integrated with an audio encoder and decoder, and may include appropriate MUX-DEMUX units, or other hardware and software, to handle encoding of both audio and video in a common data stream or separate data streams. If applicable, in some examples, MUX-DEMUX units may conform to the ITU H.223 multiplexer protocol, or other protocols such as the user datagram protocol (UDP).

[0045] The video encoder 20 and the video decoder 30 each may be implemented as any of a variety of suitable encoder 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 the video encoder 20 and the video decoder 30 may be included in one or more encoders or decoders, either of which may be integrated as part of a combined encoder/decoder (CODEC) in a respective device.

Video Coding Process

[0046] As mentioned briefly above, video encoder 20 encodes video data. The video data may comprise one or more pictures. Each of the pictures is a still image forming part of a video. In some instances, a picture may be referred to as a video “frame.” When video encoder 20 encodes the video data, video encoder 20 may generate a bitstream. The bitstream may include a sequence of bits that form a coded representation of the video data. The bitstream may include coded pictures and associated data. A coded picture is a coded representation of a picture.

[0047] To generate the bitstream, video encoder 20 may perform encoding operations on each picture in the video data. When video encoder 20 performs encoding operations on the pictures, video encoder 20 may generate a series of coded pictures and associated data. The associated data may include video parameter sets (VPS), sequence parameter sets, picture parameter sets, adaptation parameter sets, and other syntax structures. A sequence parameter set (SPS) may contain parameters applicable to zero or more sequences of pictures. A picture parameter set (PPS) may contain parameters applicable to zero or more pictures. An adaptation parameter set (APS) may contain parameters applicable to zero or more pictures. Parameters in an APS may be parameters that are more likely to change than parameters in a PPS.

[0048] To generate a coded picture, video encoder 20 may partition a picture into equally-sized video blocks. A video block may be a two-dimensional array of samples. Each of the video blocks is associated with a treeblock. In some instances, a treeblock may be referred to as a largest coding unit (LCU). The treeblocks of HEVC may be broadly analogous to the macroblocks of previous standards, such as H.264/AVC. However, a treeblock is not necessarily limited to a particular size and may include one or more coding units (CUs). Video encoder 20 may use quadtree partitioning to partition the video blocks of treeblocks into video blocks associated with CUs, hence the name "treeblocks."

[0049] In some examples, video encoder 20 may partition a picture into a plurality of slices. Each of the slices may include an integer number of CUs. In some instances, a slice comprises an integer number of treeblocks. In other instances, a boundary of a slice may be within a treeblock.

[0050] As part of performing an encoding operation on a picture, video encoder 20 may perform encoding operations on each slice of the picture. When video encoder 20 performs an encoding operation on a slice, video encoder 20 may generate encoded data associated with the slice. The encoded data associated with the slice may be referred to as a "coded slice."

[0051] To generate a coded slice, video encoder 20 may perform encoding operations on each treeblock in a slice. When video encoder 20 performs an encoding operation on a treeblock, video encoder 20 may generate a coded treeblock. The coded treeblock may comprise data representing an encoded version of the treeblock.

[0052] When video encoder 20 generates a coded slice, video encoder 20 may perform encoding operations on (e.g., encode) the treeblocks in the slice according to a raster scan order. For example, video encoder 20 may encode the treeblocks of the slice in an order that proceeds from left to right across a topmost row of treeblocks in the slice, then from left to right across a next lower row of treeblocks, and so on until video encoder 20 has encoded each of the treeblocks in the slice.

[0053] As a result of encoding the treeblocks according to the raster scan order, the treeblocks above and to the left of a given treeblock may have been encoded, but treeblocks below and to the right of the given treeblock have not yet been encoded. Consequently, video encoder 20 may be able to access information generated by encoding treeblocks above and to the left of the given treeblock when encoding the given treeblock. However, video encoder 20 may be unable to access information generated by encoding treeblocks below and to the right of the given treeblock when encoding the given treeblock.

[0054] To generate a coded treeblock, video encoder 20 may recursively perform quadtree partitioning on the video block of the treeblock to divide the video block into progressively smaller video blocks. Each of the smaller video blocks may be associated with a different CU. For example, video encoder 20 may partition the video block of a treeblock into four equally-sized sub-blocks, partition one or more of the sub-blocks into four equally-sized sub-sub-blocks, and so on. A partitioned CU may be a CU whose video block is partitioned into video blocks associated with other CUs. A non-partitioned CU may be a CU whose video block is not partitioned into video blocks associated with other CUs.

[0055] One or more syntax elements in the bitstream may indicate a maximum number of times video encoder 20 may partition the video block of a treeblock. A video block of a CU may be square in shape. The size of the video block of a CU (e.g., the size of the CU) may range from 8x8 pixels up to the size of a video block of a treeblock (e.g., the size of the treeblock) with a maximum of 64x64 pixels or greater.

[0056] Video encoder 20 may perform encoding operations on (e.g., encode) each CU of a treeblock according to a z-scan order. In other words, video encoder 20 may encode a top-left CU, a top-right CU, a bottom-left CU, and then a bottom-right CU, in that order. When video encoder 20 performs an encoding operation on a partitioned CU, video encoder 20 may encode CUs associated with sub-blocks of the video block of the partitioned CU according to the z-scan order. In other words, video encoder 20 may encode a CU associated with a top-left sub-block, a CU associated with a top-right sub-block, a CU associated with a bottom-left sub-block, and then a CU associated with a bottom-right sub-block, in that order.

[0057] As a result of encoding the CUs of a treeblock according to a z-scan order, the CUs above, above-and-to-the-left, above-and-to-the-right, left, and below-and-to-the left of a given CU may have been encoded. CUs below and to the right of the given CU have not yet been encoded. Consequently, video encoder 20 may be able to access information generated by encoding some CUs that neighbor the given CU when encoding the given CU. However, video encoder 20 may be unable to access information generated by encoding other CUs that neighbor the given CU when encoding the given CU.

[0058] When video encoder 20 encodes a non-partitioned CU, video encoder 20 may generate one or more prediction units (PUs) for the CU. Each of the PUs of the CU may be associated with a different video block within the video block of the CU. Video encoder 20 may generate a predicted video block for each PU of the CU. The predicted video block of a PU may be a block of samples. Video encoder 20 may use intra prediction or inter prediction to generate the predicted video block for a PU.

[0059] When video encoder 20 uses intra prediction to generate the predicted video block of a PU, video encoder 20 may generate the predicted video block of the PU based on decoded samples of the picture associated with the PU. If video encoder 20 uses intra prediction to generate predicted video blocks of the PUs of a CU, the CU is an intra-predicted CU. When video encoder 20 uses inter prediction to generate the predicted video block of the PU, video encoder 20 may generate the predicted video block of the PU based on decoded samples of one or more pictures other than the picture associated with the PU. If video encoder 20 uses inter prediction to generate predicted video blocks of the PUs of a CU, the CU is an inter-predicted CU.

[0060] Furthermore, when video encoder 20 uses inter prediction to generate a predicted video block for a PU, video encoder 20 may generate motion information for the PU. The motion information for a PU may indicate one or more reference blocks of the PU. Each reference block of the PU may be a video block within a reference picture. The reference picture may be a picture other than the picture associated with the PU. In some instances, a reference block of a PU may also be referred to as the "reference sample" of the PU. Video encoder 20 may generate the predicted video block for the PU based on the reference blocks of the PU.

[0061] After video encoder 20 generates predicted video blocks for one or more PUs of a CU, video encoder 20 may generate residual data for the CU based on the predicted video blocks for the PUs of the CU. The residual data for the CU may indicate differences between samples in the predicted video blocks for the PUs of the CU and the original video block of the CU.

[0062] Furthermore, as part of performing an encoding operation on a non-partitioned CU, video encoder 20 may perform recursive quadtree partitioning on the residual data of the CU to partition the residual data of the CU into one or more blocks of residual data (e.g., residual video blocks) associated with transform units (TUs) of the CU. Each TU of a CU may be associated with a different residual video block.

[0063] Video encoder 20 may apply one or more transforms to residual video blocks associated with the TUs to generate transform coefficient blocks (e.g., blocks of transform coefficients) associated with the TUs. Conceptually, a transform coefficient block may be a two-dimensional (2D) matrix of transform coefficients.

[0064] After generating a transform coefficient block, video encoder 20 may perform a quantization process on the transform 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. 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.

[0065] Video encoder 20 may associate each CU with a quantization parameter (QP) value. The QP value associated with a CU may determine how video encoder 20 quantizes transform coefficient blocks associated with the CU. Video encoder 20 may adjust the degree of quantization applied to the transform coefficient blocks associated with a CU by adjusting the QP value associated with the CU.

[0066] After video encoder 20 quantizes a transform coefficient block, video encoder 20 may generate sets of syntax elements that represent the transform coefficients in the quantized transform coefficient block. Video encoder 20 may apply entropy encoding operations, such as Context Adaptive Binary Arithmetic Coding (CABAC) operations, to some of these syntax elements. Other entropy coding techniques such as content adaptive variable length coding (CAVLC), probability interval partitioning entropy (PIPE) coding, or other binary arithmetic coding could also be used.

[0067] The bitstream generated by video encoder 20 may include a series of Network Abstraction Layer (NAL) units. Each of the NAL units may be a syntax structure containing an indication of a type of data in the NAL unit and bytes containing the data. For example, a NAL unit may contain data representing a video parameter set, a sequence parameter set, a picture parameter set, a coded slice, supplemental enhancement information (SEI), an access unit delimiter, filler data, or another type of data. The data in a NAL unit may include various syntax structures.

[0068] Video decoder 30 may receive the bitstream generated by video encoder 20. The bitstream may include a coded representation of the video data encoded by video encoder

20. When video decoder 30 receives the bitstream, video decoder 30 may perform a parsing operation on the bitstream. When video decoder 30 performs the parsing operation, video decoder 30 may extract syntax elements from the bitstream. Video decoder 30 may reconstruct the pictures of the video data based on the syntax elements extracted from the bitstream. The process to reconstruct the video data based on the syntax elements may be generally reciprocal to the process performed by video encoder 20 to generate the syntax elements.

[0069] After video decoder 30 extracts the syntax elements associated with a CU, video decoder 30 may generate predicted video blocks for the PUs of the CU based on the syntax elements. In addition, video decoder 30 may inverse quantize transform coefficient blocks associated with TUs of the CU. Video decoder 30 may perform inverse transforms on the transform coefficient blocks to reconstruct residual video blocks associated with the TUs of the CU. After generating the predicted video blocks and reconstructing the residual video blocks, video decoder 30 may reconstruct the video block of the CU based on the predicted video blocks and the residual video blocks. In this way, video decoder 30 may reconstruct the video blocks of CUs based on the syntax elements in the bitstream.

<u>Video Encoder</u>

[0070] **FIG. 2A** is a block diagram illustrating an example of a video encoder that may implement techniques in accordance with aspects described in this disclosure. Video encoder 20 may be configured to process a single layer of a video frame, such as for HEVC. Further, video encoder 20 may be configured to perform any or all of the techniques of this disclosure. As one example, prediction processing unit 100 may be configured to perform any or all of the techniques described in this disclosure. In another embodiment, the video encoder 20 includes an optional inter-layer prediction unit 128 that is configured to perform any or all of the techniques described in this disclosure. In other embodiments, inter-layer prediction can be performed by prediction processing unit 100 (e.g., inter prediction unit 121 and/or intra prediction unit 126), in which case the inter-layer prediction unit 128 may be omitted. However, aspects of this disclosure are not so limited. In some examples, the techniques described in this disclosure may be shared among the various components of video encoder 20. In some examples, additionally or alternatively, a processor (not shown) may be configured to perform any or all of the techniques described in this disclosure.

[0071] For purposes of explanation, this disclosure describes video encoder 20 in the context of HEVC coding. However, the techniques of this disclosure may be applicable to other coding standards or methods. The example depicted in **FIG. 2A** is for a single layer codec. However, as will be described further with respect to **FIG. 2B**, some or all of the video encoder 20 may be duplicated for processing of a multi-layer codec.

[0072] Video encoder 20 may perform intra- and inter-coding of video blocks within video slices. Intra coding relies on spatial prediction to reduce or remove spatial redundancy in video within a given video frame or picture. Inter-coding relies on temporal prediction to reduce or remove temporal redundancy in video within adjacent frames or pictures of a video sequence. Intra-mode (I mode) may refer to any of several spatial based coding modes. Inter-modes, such as uni-directional prediction (P mode) or bi-directional prediction (B mode), may refer to any of several temporal-based coding modes.

[0073] In the example of **FIG. 2A**, video encoder 20 includes a plurality of functional components. The functional components of video encoder 20 include a prediction processing unit 100, a residual generation unit 102, a transform processing unit 104, a quantization unit 106, an inverse quantization unit 108, an inverse transform unit 110, a reconstruction unit 112, a filter unit 113, a decoded picture buffer 114, and an entropy encoding unit 116. Prediction processing unit 100 includes an inter prediction unit 121, a motion estimation unit 122, a motion compensation unit 124, an intra prediction unit 126, and an inter-layer prediction unit 128. In other examples, video encoder 20 may include more, fewer, or different functional components. Furthermore, motion estimation unit 122 and motion compensation unit 124 may be highly integrated, but are represented in the example of **FIG. 2A** separately for purposes of explanation.

[0074] Video encoder 20 may receive video data. Video encoder 20 may receive the video data from various sources. For example, video encoder 20 may receive the video data from video source 18 (e.g., shown in **FIG. 1A** or **1B**) or another source. The video data may represent a series of pictures. To encode the video data, video encoder 20 may perform an encoding operation on each of the pictures. As part of performing the encoding operation on a picture, video encoder 20 may perform encoding operations on each slice of the picture. As part of performing an encoding operation on a slice, video encoder 20 may perform encoding operations on treeblocks in the slice.

[0075] As part of performing an encoding operation on a treeblock, prediction processing unit 100 may perform quadtree partitioning on the video block of the treeblock to divide the video block into progressively smaller video blocks. Each of the smaller video blocks may be associated with a different CU. For example, prediction processing unit 100 may partition a video block of a treeblock into four equally-sized sub-blocks, partition one or more of the sub-blocks into four equally-sized sub-sub-blocks, and so on.

[0076] The sizes of the video blocks associated with CUs may range from 8x8 samples up to the size of the treeblock with a maximum of 64x64 samples or greater. In this disclosure, "NxN" and "N by N" may be used interchangeably to refer to the sample dimensions of a video block in terms of vertical and horizontal dimensions, e.g., 16x16 samples or 16 by 16 samples. In general, a 16x16 video block has sixteen samples in a vertical direction ($y = 16$) and sixteen samples in a horizontal direction ($x = 16$). Likewise, an NxN block generally has N samples in a vertical direction and N samples in a horizontal direction, where N represents a nonnegative integer value.

[0077] Furthermore, as part of performing the encoding operation on a treeblock, prediction processing unit 100 may generate a hierarchical quadtree data structure for the treeblock. For example, a treeblock may correspond to a root node of the quadtree data structure. If prediction processing unit 100 partitions the video block of the treeblock into four sub-blocks, the root node has four child nodes in the quadtree data structure. Each of the child nodes corresponds to a CU associated with one of the sub-blocks. If prediction processing unit 100 partitions one of the sub-blocks into four sub-sub-blocks, the node corresponding to the CU associated with the sub-block may have four child nodes, each of which corresponds to a CU associated with one of the sub-sub-blocks.

[0078] Each node of the quadtree data structure may contain syntax data (e.g., syntax elements) for the corresponding treeblock or CU. For example, a node in the quadtree may include a split flag that indicates whether the video block of the CU corresponding to the node is partitioned (e.g., split) into four sub-blocks. Syntax elements for a CU may be defined recursively, and may depend on whether the video block of the CU is split into sub-blocks. A CU whose video block is not partitioned may correspond to a leaf node in the quadtree data structure. A coded treeblock may include data based on the quadtree data structure for a corresponding treeblock.

[0079] Video encoder 20 may perform encoding operations on each non-partitioned CU of a treeblock. When video encoder 20 performs an encoding operation on a non-partitioned CU, video encoder 20 generates data representing an encoded representation of the non-partitioned CU.

[0080] As part of performing an encoding operation on a CU, prediction processing unit 100 may partition the video block of the CU among one or more PUs of the CU. Video encoder 20 and video decoder 30 may support various PU sizes. Assuming that the size of a particular CU is 2Nx2N, video encoder 20 and video decoder 30 may support PU sizes of 2Nx2N or NxN, and inter-prediction in symmetric PU sizes of 2Nx2N, 2NxN, Nx2N, NxN, 2NxnU, nLx2N, nRx2N, or similar. Video encoder 20 and video decoder 30 may also support asymmetric partitioning for PU sizes of 2NxnU, 2NxnD, nLx2N, and nRx2N. In some examples, prediction processing unit 100 may perform geometric partitioning to partition the video block of a CU among PUs of the CU along a boundary that does not meet the sides of the video block of the CU at right angles.

[0081] Inter prediction unit 121 may perform inter prediction on each PU of the CU. Inter prediction may provide temporal compression. To perform inter prediction on a PU, motion estimation unit 122 may generate motion information for the PU. Motion compensation unit 124 may generate a predicted video block for the PU based the motion information and decoded samples of pictures other than the picture associated with the CU (e.g., reference pictures). In this disclosure, a predicted video block generated by motion compensation unit 124 may be referred to as an inter-predicted video block.

[0082] Slices may be I slices, P slices, or B slices. Motion estimation unit 122 and motion compensation unit 124 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, motion estimation unit 122 and motion compensation unit 124 do not perform inter prediction on the PU.

[0083] If the PU is in a P slice, the picture containing the PU is associated with a list of reference pictures referred to as “list 0.” Each of the reference pictures in list 0 contains samples that may be used for inter prediction of other pictures. When motion estimation unit 122 performs the motion estimation operation with regard to a PU in a P slice, motion estimation unit 122 may search the reference pictures in list 0 for a reference block for the PU. The reference block of the PU may be a set of samples, e.g., a block of samples, that most closely corresponds to the samples in the video block of the PU. Motion estimation unit 122 may use a variety of metrics to determine how closely a set of samples in a reference picture corresponds to the samples in the video block of a PU. For example, motion estimation unit 122 may determine how closely a set of samples in a reference picture corresponds to the samples in the video block of a PU by sum of absolute difference (SAD), sum of square difference (SSD), or other difference metrics.

[0084] After identifying a reference block of a PU in a P slice, motion estimation unit 122 may generate a reference index that indicates the reference picture in list 0 containing the reference block and a motion vector that indicates a spatial displacement between the PU and the reference block. In various examples, motion estimation unit 122 may generate motion vectors to varying degrees of precision. For example, motion estimation unit 122 may generate motion vectors at one-quarter sample precision, one-eighth sample precision, or other fractional sample precision. In the case of fractional sample precision, reference block values may be interpolated from integer-position sample values in the reference picture. Motion estimation unit 122 may output the reference index and the motion vector as the motion information of the PU. Motion compensation unit 124 may generate a predicted video block of the PU based on the reference block identified by the motion information of the PU.

[0085] If the PU is in a B slice, the picture containing the PU may be associated with two lists of reference pictures, referred to as "list 0" and "list 1." In some examples, a picture containing a B slice may be associated with a list combination that is a combination of list 0 and list 1.

[0086] Furthermore, if the PU is in a B slice, motion estimation unit 122 may perform uni-directional prediction or bi-directional prediction for the PU. When motion estimation unit 122 performs uni-directional prediction for the PU, motion estimation unit 122 may search the reference pictures of list 0 or list 1 for a reference block for the PU. Motion estimation unit 122 may then generate a reference index that indicates the reference picture in list 0 or list 1 that contains the reference block and a motion vector that indicates a spatial displacement between the PU and the reference block. Motion estimation unit 122 may output the reference index, a prediction direction indicator, and the motion vector as the motion information of the PU. The prediction direction indicator may indicate whether the reference index indicates a reference picture in list 0 or list 1. Motion compensation unit 124 may generate the predicted video block of the PU based on the reference block indicated by the motion information of the PU.

[0087] When motion estimation unit 122 performs bi-directional prediction for a PU, motion estimation unit 122 may search the reference pictures in list 0 for a reference block for the PU and may also search the reference pictures in list 1 for another reference block for the PU. Motion estimation unit 122 may then generate reference indexes that indicate the reference pictures in list 0 and list 1 containing the reference blocks and motion vectors that indicate spatial displacements between the reference blocks and the PU. Motion estimation unit 122 may output the reference indexes and the motion vectors of the PU as the motion information of the PU. Motion compensation unit 124 may generate the predicted video block of the PU based on the reference blocks indicated by the motion information of the PU.

[0088] In some instances, motion estimation unit 122 does not output a full set of motion information for a PU to entropy encoding unit 116. Rather, motion estimation unit 122 may signal the motion information of a PU with reference to the motion information of another PU. For example, motion estimation unit 122 may determine that the motion information of the PU is sufficiently similar to the motion information of a neighboring PU. In this example, motion estimation unit 122 may indicate, in a syntax structure associated with the PU, a value that indicates to video decoder 30 that the PU has the same motion information as the neighboring PU. In another example, motion estimation unit 122 may identify, in a syntax structure associated with the PU, a neighboring PU and a motion vector difference (MVD). The motion vector difference indicates a difference between the motion vector of the PU and the motion vector of the indicated neighboring PU. Video decoder 30 may use the motion vector of the indicated neighboring PU and the motion vector difference to determine the motion vector of the PU. By referring to the motion information of a first PU when signaling the motion information of a second PU, video encoder 20 may be able to signal the motion information of the second PU using fewer bits.

[0089] As further discussed below with reference to **FIGS. 4** and **5**, the prediction processing unit 100 may be configured to code (e.g., encode or decode) the PU (or any other reference layer and/or enhancement layer blocks or video units) by performing the methods illustrated in **FIGS. 4** and **5**. For example, inter prediction unit 121 (e.g., via motion estimation unit 122 and/or motion compensation unit 124), intra prediction unit 126, or inter-layer prediction unit 128 may be configured to perform the methods illustrated in **FIGS. 4** and **5**, either together or separately.

[0090] As part of performing an encoding operation on a CU, intra prediction unit 126 may perform intra prediction on PUs of the CU. Intra prediction may provide spatial compression. When intra prediction unit 126 performs intra prediction on a PU, intra prediction unit 126 may generate prediction data for the PU based on decoded samples of other PUs in the same picture. The prediction data for the PU may include a predicted video block and various syntax elements. Intra prediction unit 126 may perform intra prediction on PUs in I slices, P slices, and B slices.

[0091] To perform intra prediction on a PU, intra prediction unit 126 may use multiple intra prediction modes to generate multiple sets of prediction data for the PU. When intra prediction unit 126 uses an intra prediction mode to generate a set of prediction data for the PU, intra prediction unit 126 may extend samples from video blocks of neighboring PUs across the video block of the PU in a direction and/or gradient associated with the intra prediction mode. 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 treeblocks. Intra prediction unit 126 may use various numbers of intra prediction modes, e.g., 33 directional intra prediction modes, depending on the size of the PU.

[0092] Prediction processing unit 100 may select the prediction data for a PU from among the prediction data generated by motion compensation unit 124 for the PU or the prediction data generated by intra prediction unit 126 for the PU. In some examples, prediction processing unit 100 selects the prediction data for the PU based on rate/distortion metrics of the sets of prediction data.

[0093] If prediction processing unit 100 selects prediction data generated by intra prediction unit 126, prediction processing unit 100 may signal the intra prediction mode that was used to generate the prediction data for the PUs, e.g., the selected intra prediction mode. Prediction processing unit 100 may signal the selected intra prediction mode in various ways. For example, it is probable the selected intra prediction mode is the same as the intra prediction mode of a neighboring PU. In other words, the intra prediction mode of the neighboring PU may be the most probable mode for the current PU. Thus, prediction processing unit 100 may generate a syntax element to indicate that the selected intra prediction mode is the same as the intra prediction mode of the neighboring PU.

[0094] As discussed above, the video encoder 20 may include inter-layer prediction unit 128. Inter-layer prediction unit 128 is configured to predict a current block (e.g., a current block in the EL) using one or more different layers that are available in SVC (e.g., a base or reference layer). Such prediction may be referred to as inter-layer prediction. Inter-layer prediction unit 128 utilizes prediction methods to reduce inter-layer redundancy, thereby improving coding efficiency and reducing computational resource requirements. Some examples of inter-layer prediction include inter-layer intra prediction, inter-layer motion prediction, and inter-layer residual prediction. Inter-layer intra prediction uses the reconstruction of co-located blocks in the base layer to predict the current block in the enhancement layer. Inter-layer motion prediction uses motion information of the base layer to predict motion in the enhancement layer. Inter-layer residual prediction uses the residue of the base layer to predict the residue of the enhancement layer. Each of the inter-layer prediction schemes is discussed below in greater detail.

[0095] After prediction processing unit 100 selects the prediction data for PUs of a CU, residual generation unit 102 may generate residual data for the CU by subtracting (e.g., indicated by the minus sign) the predicted video blocks of the PUs of the CU from the video block of the CU. The residual data of a CU may include 2D residual video blocks that correspond to different sample components of the samples in the video block of the CU. For example, the residual data may include a residual video block that corresponds to differences between luminance components of samples in the predicted video blocks of the PUs of the CU and luminance components of samples in the original video block of the CU. In addition, the residual data of the CU may include residual video blocks that correspond to the differences between chrominance components of samples in the predicted video blocks of the PUs of the CU and the chrominance components of the samples in the original video block of the CU.

[0096] Prediction processing unit 100 may perform quadtree partitioning to partition the residual video blocks of a CU into sub-blocks. Each undivided residual video block may be associated with a different TU of the CU. The sizes and positions of the residual video blocks associated with TUs of a CU may or may not be based on the sizes and positions of video blocks associated with the PUs of the CU. A quadtree structure known as a "residual quad tree" (RQT) may include nodes associated with each of the residual video blocks. The TUs of a CU may correspond to leaf nodes of the RQT.

[0097] Transform processing unit 104 may generate one or more transform coefficient blocks for each TU of a CU by applying one or more transforms to a residual video block associated with the TU. Each of the transform coefficient blocks may be a 2D matrix of transform coefficients. Transform processing unit 104 may apply various transforms to the residual video 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 the residual video block associated with a TU.

[0098] After transform processing unit 104 generates a transform coefficient block associated with a TU, quantization unit 106 may quantize the transform coefficients in the transform coefficient block. Quantization unit 106 may quantize a transform coefficient block associated with a TU of a CU based on a QP value associated with the CU.

[0099] Video encoder 20 may associate a QP value with a CU in various ways. For example, video encoder 20 may perform a rate-distortion analysis on a treeblock associated with the CU. In the rate-distortion analysis, video encoder 20 may generate multiple coded representations of the treeblock by performing an encoding operation multiple times on the treeblock. Video encoder 20 may associate different QP values with the CU when video encoder 20 generates different encoded representations of the treeblock. Video encoder 20 may signal that a given QP value is associated with the CU when the given QP value is associated with the CU in a coded representation of the treeblock that has a lowest bitrate and distortion metric.

[00100] Inverse quantization unit 108 and inverse transform unit 110 may apply inverse quantization and inverse transforms to the transform coefficient block, respectively, to reconstruct a residual video block from the transform coefficient block. Reconstruction unit 112 may add the reconstructed residual video block to corresponding samples from one or more predicted video blocks generated by prediction processing unit 100 to produce a reconstructed video block associated with a TU. By reconstructing video blocks for each TU of a CU in this way, video encoder 20 may reconstruct the video block of the CU.

[00101] After reconstruction unit 112 reconstructs the video block of a CU, filter unit 113 may perform a deblocking operation to reduce blocking artifacts in the video block associated with the CU. After performing the one or more deblocking operations, filter unit 113 may store the reconstructed video block of the CU in decoded picture buffer 114. Motion estimation unit 122 and motion compensation unit 124 may use a reference picture that contains the reconstructed video block to perform inter prediction on PUs of subsequent pictures. In addition, intra prediction unit 126 may use reconstructed video blocks in decoded picture buffer 114 to perform intra prediction on other PUs in the same picture as the CU.

[00102] Entropy encoding unit 116 may receive data from other functional components of video encoder 20. For example, entropy encoding unit 116 may receive transform coefficient blocks from quantization unit 106 and may receive syntax elements from prediction processing unit 100. When entropy encoding unit 116 receives the data, entropy encoding unit 116 may perform one or more entropy encoding operations to generate entropy encoded data. For example, video encoder 20 may perform a context adaptive variable length coding (CAVLC) operation, a CABAC 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, or another type of entropy encoding operation on the data. Entropy encoding unit 116 may output a bitstream that includes the entropy encoded data.

[00103] As part of performing an entropy encoding operation on data, entropy encoding unit 116 may select a context model. If entropy encoding unit 116 is performing a CABAC operation, the context model may indicate estimates of probabilities of particular bins having particular values. In the context of CABAC, the term “bin” is used to refer to a bit of a binarized version of a syntax element.

Multi-Layer Video Encoder

[00104] **FIG. 2B** is a block diagram illustrating an example of a multi-layer video encoder 23 that may implement techniques in accordance with aspects described in this disclosure. The video encoder 23 may be configured to process multi-layer video frames, such as for SHVC and multiview coding. Further, the video encoder 23 may be configured to perform any or all of the techniques of this disclosure.

[00105] The video encoder 23 includes a video encoder 20A and video encoder 20B, each of which may be configured as the video encoder 20 and may perform the functions described above with respect to the video encoder 20. Further, as indicated by the reuse of reference numbers, the video encoders 20A and 20B may include at least some of the systems and subsystems as the video encoder 20. Although the video encoder 23 is illustrated as including two video encoders 20A and 20B, the video encoder 23 is not limited as such and may include any number of video encoder 20 layers. In some embodiments, the video encoder 23 may include a video encoder 20 for each picture or frame in an access unit. For example, an access unit that includes five pictures may be processed or encoded by a video encoder that includes five encoder layers. In some embodiments, the video encoder 23 may include more encoder layers than frames in an access unit. In some such cases, some of the video encoder layers may be inactive when processing some access units.

[00106] In addition to the video encoders 20A and 20B, the video encoder 23 may include an resampling unit 90. The resampling unit 90 may, in some cases, upsample a base layer of a received video frame to, for example, create an enhancement layer. The resampling unit 90 may upsample particular information associated with the received base layer of a frame, but not other information. For example, the resampling unit 90 may upsample the spatial size or number of pixels of the base layer, but the number of slices or the picture order count may remain constant. In some cases, the resampling unit 90 may not process the received video and/or may be optional. For example, in some cases, the prediction processing unit 100 may perform upsampling. In some embodiments, the resampling unit 90 is configured to upsample a layer and reorganize, redefine, modify, or adjust one or more slices to comply with a set of slice boundary rules and/or raster scan rules. Although primarily described as upsampling a base layer, or a lower layer in an access unit, in some cases, the resampling unit 90 may downsample a layer. For example, if during streaming of a video bandwidth is reduced, a frame may be downsampled instead of upsampled.

[00107] The resampling unit 90 may be configured to receive a picture or frame (or picture information associated with the picture) from the decoded picture buffer 114 of the lower layer encoder (e.g., the video encoder 20A) and to upsample the picture (or the received picture information). This upsampled picture may then be provided to the prediction processing unit 100 of a higher layer encoder (e.g., the video encoder 20B) configured to encode a picture in the same access unit as the lower layer encoder. In some cases, the higher layer encoder is one layer removed from the lower layer encoder. In other cases, there may be one or more higher layer encoders between the layer 0 video encoder and the layer 1 encoder of **FIG. 2B**.

[00108] In some cases, the resampling unit 90 may be omitted or bypassed. In such cases, the picture from the decoded picture buffer 114 of the video encoder 20A may be provided directly, or at least without being provided to the resampling unit 90, to the prediction processing unit 100 of the video encoder 20B. For example, if video data provided to the video encoder 20B and the reference picture from the decoded picture buffer 114 of the video encoder 20A are of the same size or resolution, the reference picture may be provided to the video encoder 20B without any resampling.

[00109] In some embodiments, the video encoder 23 downsamples video data to be provided to the lower layer encoder using the downsampling unit 94 before provided the video data to the video encoder 20A. Alternatively, the downsampling unit 94 may be a resampling unit 90 capable of upsampling or downsampling the video data. In yet other embodiments, the downsampling unit 94 may be omitted.

[00110] As illustrated in **FIG. 2B**, the video encoder 23 may further include a multiplexor 98, or mux. The mux 98 can output a combined bitstream from the video encoder 23. The combined bitstream may be created by taking a bitstream from each of the video encoders 20A and 20B and alternating which bitstream is output at a given time. While in some cases the bits from the two (or more in the case of more than two video encoder layers) bitstreams may be alternated one bit at a time, in many cases the bitstreams are combined differently. For example, the output bitstream may be created by alternating the selected bitstream one block at a time. In another example, the output bitstream may be created by outputting a non-1:1 ratio of blocks from each of the video encoders 20A and 20B. For instance, two blocks may be output from the video encoder 20B for each block output from the video encoder 20A. In some embodiments, the output stream from the mux 98 may be preprogrammed. In other embodiments, the mux 98 may combine the bitstreams from the video encoders 20A, 20B based on a control signal received from a system external to the video encoder 23, such as from a processor on a source device including the source module 12. The control signal may be generated based on the resolution or bitrate of a video from the video source 18, based on a bandwidth of the link 16, based on a subscription associated with a user (e.g., a paid subscription versus a free subscription), or based on any other factor for determining a resolution output desired from the video encoder 23.

Video Decoder

[00111] **FIG. 3A** is a block diagram illustrating an example of a video decoder that may implement techniques in accordance with aspects described in this disclosure. The video decoder 30 may be configured to process a single layer of a video frame, such as for HEVC. Further, video decoder 30 may be configured to perform any or all of the techniques of this disclosure. As one example, motion compensation unit 162 and/or intra prediction unit 164 may be configured to perform any or all of the techniques described in this disclosure. In one embodiment, video decoder 30 may optionally include inter-layer prediction unit 166 that is configured to perform any or all of the techniques described in this disclosure. In other embodiments, inter-layer prediction can be performed by prediction processing unit 152 (e.g., motion compensation unit 162 and/or intra prediction unit 164), in which case the inter-layer prediction unit 166 may be omitted. However, aspects of this disclosure are not so limited. In some examples, the techniques described in this disclosure may be shared among the various components of video decoder 30. In some examples, additionally or alternatively, a processor (not shown) may be configured to perform any or all of the techniques described in this disclosure.

[00112] 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. The example depicted in **FIG. 3A** is for a single layer codec. However, as will be described further with respect to **FIG. 3B**, some or all of the video decoder 30 may be duplicated for processing of a multi-layer codec.

[00113] In the example of **FIG. 3A**, video decoder 30 includes a plurality of functional components. The functional components of video decoder 30 include an entropy decoding unit 150, a prediction processing unit 152, an inverse quantization unit 154, an inverse transform unit 156, a reconstruction unit 158, a filter unit 159, and a decoded picture buffer 160. Prediction processing unit 152 includes a motion compensation unit 162, an intra prediction unit 164, and an inter-layer prediction unit 166. In some examples, video decoder 30 may perform a decoding pass generally reciprocal to the encoding pass described with respect to video encoder 20 of **FIG. 2A**. In other examples, video decoder 30 may include more, fewer, or different functional components.

[00114] Video decoder 30 may receive a bitstream that comprises encoded video data. The bitstream may include a plurality of syntax elements. When video decoder 30 receives the bitstream, entropy decoding unit 150 may perform a parsing operation on the bitstream. As a result of performing the parsing operation on the bitstream, entropy decoding unit 150 may extract syntax elements from the bitstream. As part of performing the parsing operation, entropy decoding unit 150 may entropy decode entropy encoded syntax elements in the bitstream. Prediction processing unit 152, inverse quantization unit 154, inverse transform unit 156, reconstruction unit 158, and filter unit 159 may perform a reconstruction operation that generates decoded video data based on the syntax elements extracted from the bitstream.

[00115] As discussed above, the bitstream may comprise a series of NAL units. The NAL units of the bitstream may include video parameter set NAL units, sequence parameter set NAL units, picture parameter set NAL units, SEI NAL units, and so on. As part of performing the parsing operation on the bitstream, entropy decoding unit 150 may perform parsing operations that extract and entropy decode sequence parameter sets from sequence parameter set NAL units, picture parameter sets from picture parameter set NAL units, SEI data from SEI NAL units, and so on.

[00116] In addition, the NAL units of the bitstream may include coded slice NAL units. As part of performing the parsing operation on the bitstream, entropy decoding unit 150 may perform parsing operations that extract and entropy decode coded slices from the coded slice NAL units. Each of the coded slices may include a slice header and slice data. The slice header may contain syntax elements pertaining to a slice. The syntax elements in the slice header may include a syntax element that identifies a picture parameter set associated with a picture that contains the slice. Entropy decoding unit 150 may perform entropy decoding operations, such as CABAC decoding operations, on syntax elements in the coded slice header to recover the slice header.

[00117] As part of extracting the slice data from coded slice NAL units, entropy decoding unit 150 may perform parsing operations that extract syntax elements from coded CUs in the slice data. The extracted syntax elements may include syntax elements associated with transform coefficient blocks. Entropy decoding unit 150 may then perform CABAC decoding operations on some of the syntax elements.

[00118] After entropy decoding unit 150 performs a parsing operation on a non-partitioned CU, video decoder 30 may perform a reconstruction operation on the non-partitioned CU. To perform the reconstruction operation on a non-partitioned 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 a residual video block associated with the CU.

[00119] As part of performing a reconstruction operation on a TU, inverse quantization unit 154 may inverse quantize, e.g., de-quantize, a transform coefficient block associated with the TU. Inverse quantization unit 154 may inverse quantize the transform coefficient block in a manner similar to the inverse quantization processes proposed for HEVC or defined by the H.264 decoding standard. Inverse quantization unit 154 may use a quantization parameter QP calculated by video encoder 20 for a CU of the transform coefficient block to determine a degree of quantization and, likewise, a degree of inverse quantization for inverse quantization unit 154 to apply.

[00120] After inverse quantization unit 154 inverse quantizes a transform coefficient block, inverse transform unit 156 may generate a residual video block for the TU associated with the transform coefficient block. Inverse transform unit 156 may apply an inverse transform to the transform coefficient block in order to generate the residual video block for the TU. For example, inverse transform 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 transform coefficient block. In some examples, inverse transform unit 156 may determine an inverse transform to apply to the transform coefficient block based on signaling from video encoder 20. In such examples, inverse transform unit 156 may determine the inverse transform based on a signaled transform at the root node of a quadtree for a treeblock associated with the transform coefficient block. In other examples, inverse transform unit 156 may infer the inverse transform from one or more coding characteristics, such as block size, coding mode, or the like. In some examples, inverse transform unit 156 may apply a cascaded inverse transform.

[00121] In some examples, motion compensation unit 162 may refine the predicted video block of a PU by performing interpolation based on interpolation filters. Identifiers for interpolation filters to be used for motion compensation with sub-sample precision may be included in the syntax elements. Motion compensation unit 162 may use the same interpolation filters used by video encoder 20 during generation of the predicted video block of the PU to calculate interpolated values for sub-integer samples of a reference block. Motion compensation unit 162 may determine the interpolation filters used by video encoder

20 according to received syntax information and use the interpolation filters to produce the predicted video block.

[00122] As further discussed below with reference to **FIGS. 4** and **5**, the prediction processing unit 152 may code (e.g., encode or decode) the PU (or any other reference layer and/or enhancement layer blocks or video units) by performing the methods illustrated in **FIGS. 4** and **5**. For example, motion compensation unit 162, intra prediction unit 164, or inter-layer prediction unit 166 may be configured to perform the methods illustrated in **FIGS. 4** and **5**, either together or separately.

[00123] If a PU is encoded using intra prediction, intra prediction unit 164 may perform intra prediction to generate a predicted video block for the PU. For example, intra prediction unit 164 may determine an intra prediction mode for the PU based on syntax elements in the bitstream. The bitstream may include syntax elements that intra prediction unit 164 may use to determine the intra prediction mode of the PU.

[00124] In some instances, the syntax elements may indicate that intra prediction unit 164 is to use the intra prediction mode of another PU to determine the intra prediction mode of the current PU. For example, it may be probable that the intra prediction mode of the current PU is the same as the intra prediction mode of a neighboring PU. In other words, the intra prediction mode of the neighboring PU may be the most probable mode for the current PU. Hence, in this example, the bitstream may include a small syntax element that indicates that the intra prediction mode of the PU is the same as the intra prediction mode of the neighboring PU. Intra prediction unit 164 may then use the intra prediction mode to generate prediction data (e.g., predicted samples) for the PU based on the video blocks of spatially neighboring PUs.

[00125] As discussed above, video decoder 30 may also include inter-layer prediction unit 166. Inter-layer prediction unit 166 is configured to predict a current block (e.g., a current block in the EL) using one or more different layers that are available in SVC (e.g., a base or reference layer). Such prediction may be referred to as inter-layer prediction. Inter-layer prediction unit 166 utilizes prediction methods to reduce inter-layer redundancy, thereby improving coding efficiency and reducing computational resource requirements. Some examples of inter-layer prediction include inter-layer intra prediction, inter-layer motion prediction, and inter-layer residual prediction. Inter-layer intra prediction uses the reconstruction of co-located blocks in the base layer to predict the current block in the enhancement layer. Inter-layer motion prediction uses motion information of the base layer to predict motion in the enhancement layer. Inter-layer residual prediction uses the residue of the base layer to predict the residue of the enhancement layer. Each of the inter-layer prediction schemes is discussed below in greater detail.

[00126] Reconstruction unit 158 may use the residual video blocks associated with TUs of a CU and the predicted video blocks of the PUs of the CU, e.g., either intra-prediction data or inter-prediction data, as applicable, to reconstruct the video block of the CU. Thus, video decoder 30 may generate a predicted video block and a residual video block based on syntax elements in the bitstream and may generate a video block based on the predicted video block and the residual video block.

[00127] After reconstruction unit 158 reconstructs the video block of the CU, filter unit 159 may perform a deblocking operation to reduce blocking artifacts associated with the CU. After filter unit 159 performs a deblocking operation to reduce blocking artifacts associated with the CU, video decoder 30 may store the video block of the CU in decoded picture buffer 160. Decoded picture buffer 160 may provide reference pictures for subsequent motion compensation, intra prediction, and presentation on a display device, such as display device 32 of **FIG. 1A** or **1B**. For instance, video decoder 30 may perform, based on the video blocks in decoded picture buffer 160, intra prediction or inter prediction operations on PUs of other CUs.

Multi-Layer Decoder

[00128] **FIG. 3B** is a block diagram illustrating an example of a multi-layer video decoder 33 that may implement techniques in accordance with aspects described in this disclosure. The video decoder 33 may be configured to process multi-layer video frames, such as for SHVC and multiview coding. Further, the video decoder 33 may be configured to perform any or all of the techniques of this disclosure.

[00129] The video decoder 33 includes a video decoder 30A and video decoder 30B, each of which may be configured as the video decoder 30 and may perform the functions described above with respect to the video decoder 30. Further, as indicated by the reuse of reference numbers, the video decoders 30A and 30B may include at least some of the systems and subsystems as the video decoder 30. Although the video decoder 33 is illustrated as including two video decoders 30A and 30B, the video decoder 33 is not limited as such and may include any number of video decoder 30 layers. In some embodiments, the video decoder 33 may include a video decoder 30 for each picture or frame in an access unit. For example, an access unit that includes five pictures may be processed or decoded by a video decoder that includes five decoder layers. In some embodiments, the video decoder 33 may include more decoder layers than frames in an access unit. In some such cases, some of the video decoder layers may be inactive when processing some access units.

[00130] In addition to the video decoders 30A and 30B, the video decoder 33 may include an upsampling unit 92. In some embodiments, the upsampling unit 92 may upsample a base layer of a received video frame to create an enhanced layer to be added to the reference picture list for the frame or access unit. This enhanced layer can be stored in the decoded picture buffer 160. In some embodiments, the upsampling unit 92 can include some or all of the embodiments described with respect to the resampling unit 90 of **FIG. 2A**. In some embodiments, the upsampling unit 92 is configured to upsample a layer and reorganize, redefine, modify, or adjust one or more slices to comply with a set of slice boundary rules and/or raster scan rules. In some cases, the upsampling unit 92 may be a resampling unit configured to upsample and/or downsample a layer of a received video frame

[00131] The upsampling unit 92 may be configured to receive a picture or frame (or picture information associated with the picture) from the decoded picture buffer 160 of the lower layer decoder (e.g., the video decoder 30A) and to upsample the picture (or the received picture information). This upsampled picture may then be provided to the prediction processing unit 152 of a higher layer decoder (e.g., the video decoder 30B) configured to decode a picture in the same access unit as the lower layer decoder. In some cases, the higher layer decoder is one layer removed from the lower layer decoder. In other cases, there may be one or more higher layer decoders between the layer 0 decoder and the layer 1 decoder of **FIG. 3B**.

[00132] In some cases, the upsampling unit 92 may be omitted or bypassed. In such cases, the picture from the decoded picture buffer 160 of the video decoder 30A may be provided directly, or at least without being provided to the upsampling unit 92, to the prediction processing unit 152 of the video decoder 30B. For example, if video data provided to the video decoder 30B and the reference picture from the decoded picture buffer 160 of the video decoder 30A are of the same size or resolution, the reference picture may be provided to the video decoder 30B without upsampling. Further, in some embodiments, the upsampling unit 92 may be a resampling unit 90 configured to upsample or downsample a reference picture received from the decoded picture buffer 160 of the video decoder 30A.

[00133] As illustrated in **FIG. 3B**, the video decoder 33 may further include a demultiplexor 99, or demux. The demux 99 can split an encoded video bitstream into multiple bitstreams with each bitstream output by the demux 99 being provided to a different video decoder 30A and 30B. The multiple bitstreams may be created by receiving a bitstream and each of the video decoders 30A and 30B receives a portion of the bitstream at a given time. While in some cases the bits from the bitstream received at the demux 99 may be alternated one bit at a time between each of the video decoders (e.g., video decoders 30A and 30B in the example of **FIG. 3B**), in many cases the bitstream is divided differently. For example, the bitstream may be divided by alternating which video decoder receives the bitstream one block at a time. In another example, the bitstream may be divided by a non-1:1 ratio of blocks to each of the video decoders 30A and 30B. For instance, two blocks may be provided to the video decoder 30B for each block provided to the video decoder 30A. In some embodiments, the division of the bitstream by the demux 99 may be preprogrammed. In other embodiments, the demux 99 may divide the bitstream based on a control signal received from a system external to the video decoder 33, such as from a processor on a destination device including the destination module 14. The control signal may be generated based on the resolution or bitrate of a video from the input interface 28, based on a bandwidth of the link 16, based on a subscription associated with a user (e.g., a paid subscription versus a free subscription), or based on any other factor for determining a resolution obtainable by the video decoder 33.

Direct Dependency Flag

[00134] In some example implementations (e.g., MV-HEVC and SHVC), there is a syntax element called direct_dependency_flag that specifies, for a particular layer, which layer or layers can be used for inter-layer prediction of the particular layer. In one embodiment, the direct_dependency_flag is a two-dimensional array that specifies whether one layer of video data is coded based on (or dependent on) another layer of video data. Such a two-dimensional array may take a form of values direct_dependency_flag[i][j], where i corresponds to the layer to be coded (e.g., current layer) and j corresponds to the layer to be referenced (e.g., reference layer). In this example, direct_dependency_flag may be 0 if the reference layer is not a direct reference layer of the current layer, and direct_dependency_flag may be 1 if the reference layer is a direct reference layer of the current layer. In one embodiment, if direct_dependency_flag is omitted or undefined, the value is inferred to be 0. In another embodiment, if direct_dependency_flag is omitted or undefined, the value is inferred to be 1. In one embodiment, if Layer A is a direct reference layer of Layer B, it means that Layer B can be coded based at least in part on information included in Layer A. In another embodiment, if Layer A is a direct reference layer of Layer B, it means that Layer B is coded based at least in part on information included in Layer A. In some embodiments, all the layers that have a smaller layer ID (e.g., lower layer) are direct reference layers of a particular layer. In other embodiments, only some of the lower layers may be direct reference layers of a particular layer. For example, the encoder may choose only some of the lower layers as direct dependency layers of a particular layer to reduce computational complexity. The applicable coding scheme (e.g., HEVC) may have a limit as to how many direct reference layers a particular layer may have (e.g., no more than one reference layer for spatial scalability). In one embodiment, the direct_dependency_flag flag is signaled in the video parameter set (VPS) and applies to the entire coded video sequence (CVS).

Direct Dependency Type

[00135] The information that is used to code the current layer may include texture information (e.g., pixel values) of the reference layer, motion information (e.g., motion vectors, reference indices, prediction direction, etc.) of the reference layer. However, the information of the reference layer that may be used to code the current layer is not limited to those discussed herein, but can be any information that is included in or part of the reference layer.

[00136] In some implementations, one or more additional flags or syntax elements may be used to indicate the type or types of information that are derived or imported from the reference layer to code the current layer. For example, in some embodiments, the reference layer may be used for inter-layer motion prediction, inter-layer texture prediction, or both. In one embodiment, such a flag or syntax element may be called "direct_dependency_type."

[00137] In one embodiment, the direct_dependency_type is a two-dimensional array that specifies which type of inter-layer prediction is used for coding the current layer using the reference layer. Such a two-dimensional array may take a form of values direct_dependency_type[i][j], where i corresponds to the current (e.g., layer to be coded) and j corresponds to the reference layer (e.g., layer to be referenced). In this example, a direct_dependency_type value of 0 may indicate inter-layer sample prediction only, 1 may indicate inter-layer motion prediction only, and 2 may indicate both inter-layer sample and motion prediction. In some embodiments, a direct_dependency_type value of 3 (or any other value) may indicate that there is no dependency. How each direct_dependency_type value is assigned or mapped to different types of inter-layer prediction may be different in other implementations, and the present disclosure is not limited to any particular assignment or mapping of direct_dependency_type values to different types of inter-layer prediction. In one embodiment, the direct_dependency_type syntax element is signaled in the video parameter set (VPS) and applies to the entire coded video sequence (CVS).

Other Information Derived From Direct Dependency Type

[00138] In some implementations, direct_dependency_type[i][j] is used to derive the variables NumSamplePredRefLayers[i], NumMotionPredRefLayers[i], SamplePredEnabledFlag[i][j], MotionPredEnabledFlag[i][j], NumDirectRefLayers[i], RefLayerId[i][j], MotionPredRefLayerId[i][j], and SamplePredRefLayerId[i][j]. In one embodiment, NumSamplePredRefLayers may indicate the number of reference layers that may be used for inter-layer sample prediction, NumMotionPredRefLayers may indicate the number of layers that may be used for inter-layer motion prediction, SamplePredEnabledFlag may indicate whether inter-layer sample prediction is enabled such that the current layer can be coded using the sample information of the reference layer, MotionPredEnabledFlag may indicate whether inter-layer sample prediction is enabled such that the current layer can be coded using the motion information of the reference layer, NumDirectRefLayers may indicate the number of direct reference layers that the current layer has, RefLayerId may indicate the layer ID of the reference layer, MotionPredRefLayerId may indicate the layer ID of the reference layer for which inter-layer motion prediction is enabled, and SamplePredRefLayerId may indicate the layer ID of the reference layer for which inter-layer sample prediction is enabled. In some embodiments, direct_dependency_type[i][j] may have values in the range of 0 to 2, inclusive, to conform to certain bitstream conformance constraints. Although the value of direct_dependency_type[i][j] may be in the range of 0 to 2, inclusive, in certain embodiments, decoders may allow values of direct_dependency_type[i][j] in the range of 3 to 2^{32}-2, inclusive, to appear in the syntax. In some implementations, although the encoder may not specify direct_dependency_type values greater than 2, the decoder may be configured to parse values greater than 2. In some implementations, direct_dependency_type values greater than 2 may indicate that there is no direct dependency. In some implementations, the length of the direct_dependency_type syntax element may be direct_dep_type_len_minus2 + 2. For example, this value is equal to 2 in some existing coding schemes (e.g., HEVC), since the length of the direct_dependency_type syntax element is 2. Other implementations, currently known or developed in the future, may use the values greater than 2 to indicate other aspects and characteristics of the coded video data.

[00139] In one example implementation, the variables NumSamplePredRefLayers[i], NumMotionPredRefLayers[i], SamplePredEnabledFlag[i][j], MotionPredEnabledFlag[i][j], NumDirectRefLayers[i], RefLayerId[i][j], MotionPredRefLayerId[i][j], and SamplePredRefLayerId[i][j] are derived as follows:

```
for( i = 0; i < 64; i++ ) {
    NumSamplePredRefLayers[ i ] = 0
    NumMotionPredRefLayers[ i ] = 0
    NumDirectRefLayers[ i ] = 0
    for( j = 0; j < 64; j++ ) {
     SamplePredEnabledFlag[ i ][ j ] = 0
     MotionPredEnabledFlag[ i ][ j ] = 0
     RefLayerId[ i ][ j ] = 0
     SamplePredRefLayerId[ i ][ j ] = 0
     MotionPredRefLayerId[ i ][ j ] = 0
    }
}
for( i = 1; i  <=  vps_max_layers_minus1; i++ ) {
    iNuhLId = layer_id_in_nuh[ i ]
    for( j = 0; j < i; j++ )
        if( direct_dependency_flag[ i ][ j ] ) {
                        RefLayerId[ iNuhLId ][ NumDirectRefLayers[ iNuhLId ]++ ] =
layer_id_in_nuh[ j ]
          SamplePredEnabledFlag[ iNuhLId ][ j ] = ( ( direct_dependency_type[ i ][ j ]
+ 1 ) & 1 )
          NumSamplePredRefLayers[ iNuhLId ] +=
SamplePredEnabledFlag[ iNuhLId ][ j ]
          MotionPredEnabledFlag[ iNuhLId ][ j ] =
( ( ( direct_dependency_type[ i ][ j ] + 1 ) & 2 ) >> 1 )
          NumMotionPredRefLayers[ iNuhLId ] +=
MotionPredEnabledFlag[ iNuhLId ][ j ]
     }
}
```

```
for( i = 1, mIdx = 0, sIdx = 0; i <= vps_max_layers_minus1; i++ ) {
   iNuhLId = layer_id_in_nuh[ i ]
   for( j = 0, j < i; j++ ) {
     if( MotionPredEnabledFlag[ iNuhLId ][ j ] )
         MotionPredRefLayerId[ iNuhLId ][ mIdx++ ] = layer_id_in_nuh[ j ]
     if( SamplePredEnabledFlag[ INuhLid ][ j ] )
         SamplePredRefLayerId[ iNuhLid ][ sIdx++ ] = layer_id_in_nuh[ j ]
   }
}
```

Reference Layer Codec

[00140] In some existing coding schemes, a reference layer codec may be HEVC or H.264/AVC, or a general, non-HEVC codec. In addition, there may be a flag in a parameter set indicating the codec to be used. For example, a flag in the video parameter set (VPS) may indicate whether HEVC or non-HEVC (e.g., AVC) codec is used to code the reference layer. In one example, a flag avc_base_layer_flag may have a value equal to 1, indicating that the reference layer codec conforms to the video coding standard according to Recommendation ITU-T H.264 | International Standard ISO/IEC 14496-10, and alternatively, may have a value equal to 0, indicating that the reference layer codec conforms to the HEVC specification. Therefore, a coding device configured to encode or decode an enhancement layer may have information regarding whether an AVC or HEVC codec (or some other non-HEVC codec) is used with respect to the reference layer.

Reference Layer Codec Type and Motion Information Availability

[00141] In some reference layer coding schemes (e.g., AVC), the motion information may not be available for coding one or more enhancement layers. For example, the reference layer coder (e.g., encoder or decoder) may only output the texture (e.g., pixel value) information (e.g., for display), and may not output the motion information that is used to code the reference layer. In such a case, the motion information may not be accessible by the enhancement layer codec and thus may not be used to code the enhancement layer.

[00142] In some embodiments, to restrict access to such unavailable information, the syntax element that may indicate the usage of such information may be restricted from providing such an indication. For example, if the syntax element is unrestricted and is allowed to indicate the usage of such information, the coder (e.g., encoder or decoder) may not be able to encode or decode a certain portion of the bitstream (e.g., if the referenced motion information is not available as described above).

[00143] By exploiting the dependence of the availability of motion information for inter-layer prediction on the coding scheme used by the reference layer, additional processing to determine the direct dependency type between certain layers may be omitted, thus resulting in improved coding efficiency and/or reduced computational complexity.

[00144] In one embodiment, the inter-layer prediction methods that may be used for coding the enhancement layer may depend on whether the reference layer codec conforms to a particular coding scheme. The particular coding scheme may be any predetermined coding scheme that may indicate whether a portion of the information in the reference layer may not be available. In one example, the particular coding scheme is AVC. In such example, the inter-layer prediction methods used for coding the enhancement layer may depend on whether the reference layer codec conforms to AVC (or a non-HEVC coding scheme). For example, the coder may check, using the codec information discussed above, which codec is used to code the reference layer, and if the reference layer codec conforms to a coding scheme other than HEVC, the coder (e.g., encoder or decoder) may disable the use of inter-layer motion prediction when coding the enhancement layer. On the other hand, if the reference layer codec conforms to HEVC, the coder (e.g., encoder or decoder) may enable the use of inter-layer motion prediction when coding the enhancement layer.

<u>**Other Information Regarding H.264/AVC Conforming Base Layer**</u>

[00145] In some implementations, the avc_base_layer_flag value of 1 indicates that when the reference layer is decoded using the Rec. ITU-T H.264 | ISO/IEC 14496-10 decoding process for reference picture list construction, the output reference picture list refPicList0 (and refPicList1, when applicable) does not contain any pictures having a temporal ID that is greater than the temporal ID of the coded picture (e.g., the picture with which the reference picture list is associated). In some implementations, all sub-bitstreams of the Rec. ITU-T H.264 | ISO/IEC 14496-10 conforming base layer that can be derived using the sub-bitstream extraction process as specified in Rec. ITUT H.264 | ISO/IEC 14496-10 Subclause G.8.8.1 with any value for temporal_id as the input shall result in a set of CVS's, with each CVS conforming to one or more of the profiles specified in Rec. ITUT H.264 | ISO/IEC 14496-10 Annexes A, G, and H.

Implementation of Codec-Dependent Signaling of Inter-Layer Prediction Type

[00146] As discussed above, the value of the flag or syntax element used to indicate the usage of the inter-layer motion prediction (or other prediction modes) can be dependent on the codec that is used to code the reference layer (or the base layer), which can be specified by, for example, avc_base_layer_flag or other syntax elements or derived from some certain syntax elements. In one embodiment, a base layer is the layer with the smallest layer ID. In another embodiment, a base layer is the layer having a layer ID of 0. In some embodiments, inter-layer motion prediction is disabled for an AVC reference layer. In other embodiments, inter-layer motion prediction is disabled for a non-HEVC reference layer. The disabling of the inter-layer motion prediction can be implemented as discussed below.

Embodiment #1

[00147] In one embodiment, direct_dependency_type takes a value equal to 0 for any non-HEVC reference layer. In another embodiment, direct_dependency_type takes a value equal to 0 for an AVC reference layer. This feature may be implemented as a bitstream constraint on the direct_dependency_type syntax element. For example, for a conforming bitstream, direct_dependency_type for a layer having a layer ID of 0 (e.g., base layer) shall equal to 0 if avc_base_layer_flag is equal to 1. In another example, a conforming bitstream shall not contain any syntax elements specifying inter-layer motion prediction for a layer having a layer ID of 0 (e.g., base layer) if avc_base_layer_flag is equal to 1.

Embodiment #2

[00148] In one embodiment, for a reference layer that is coded with a non-HEVC (e.g., AVC) coding scheme, direct_dependency_type is conditionally signaled by enabling only inter-layer sample prediction if avc_base_layer_flag is equal to 1. Since inter-layer sample prediction is the only available inter-layer prediction for non-HEVC-coded (e.g., AVC-coded) layers, direct_dependency_type signaling may be omitted and the value of direct_dependency_type can be inferred to be 0 (e.g., which indicates that only inter-layer sample prediction is used) if avc_base_layer_flag is equal to 1.

[00149] Shown below is an example syntax, where the omission of the direct_dependency_type signaling is implemented in the portion that is *italicized*.

vps_extension( ) {	Descriptor
…	
avc_base_layer_flag	u(1)
…	
for(i = 1; i <= vps_max_layers_minus1; i++)	
for(j = (*avc_base_layer_flag ? 1 : 0*); j < i; j++)	
if(direct_dependency_flag[i][j])	
direct_dependency_type[i][j]	u(v)
…	
}	

Table 1. Example Syntax #1

[00150] In some implementations, the inferring of direct_dependency_type (e.g., when a value of 0 is to be inferred) is added to the semantics of direct_dependency_type. For example, if direct_dependency_type is not present for a certain layer, direct_dependency_type is inferred to be 0 for that layer. Alternatively, if direct_dependency_type is not present for a certain layer, direct_dependency_type is inferred to be 0 for that layer if avc_base_layer_flag is equal to 1. The certain layer discussed above may be a base layer having a layer ID of 0. In another embodiment, the certain layer may be a reference layer having a non-zero layer ID.

Other Syntax Elements

[00151] Although in the present disclosure, the direct_dependency_type syntax element that indicates the type of inter-layer prediction is used to describe various embodiments, the same techniques and mechanisms can be applied and extended to other syntax elements that can specify the inter-layer prediction types.

[00152] In the present disclosure, the avc_base_layer_flag syntax element is used to indicate the codec scheme of the base layer, or to indicate that the base layer is coded with non-HEVC codec scheme. However, other syntax elements and mechanisms can be signalled or processed to specify the reference layer or base layer codec scheme.

[00153] The techniques and mechanisms described herein are not limited to signaling the availability or unavailability of motion information, and other techniques and mechanisms similar to those described in the present disclosure can be used to indicate the availability or unavailability of texture information or other types of information for inter-layer prediction (or prediction in general).

Example Flowchart #1

[00154] **FIG. 4** is a flowchart illustrating a method 400 for coding video information, according to an embodiment of the present disclosure. The steps illustrated in **FIG. 4** may be performed by an encoder (e.g., the video encoder as shown in **FIG. 2A** or **FIG. 2B**), a decoder (e.g., the video decoder as shown in **FIG. 3A** or **FIG. 3B**), or any other component. For convenience, method 400 is described as performed by a coder, which may be the encoder, the decoder, or another component.

[00155] The method 400 begins at block 401. In block 405, the coder determines whether a reference layer codec is a particular type of codec. In one embodiment, the reference layer codec specifies the type of codec that is used to code the reference layer. In another embodiment, the reference layer codec specifies the type of coding scheme to which it conforms. In one example, the particular type of codec includes an AVC codec. In another example, the particular type of codec includes a non-HEVC codec. If the coder determines that the reference layer codec is not the particular type of codec, the coder enables inter-layer motion prediction in block 410. In one example, enabling inter-layer motion prediction may comprise actually coding the current layer using the motion information of the reference layer. In another example, enabling inter-layer motion prediction may comprise refraining from disabling inter-layer motion prediction and may not mean that motion information of the reference layer is actually used to code the current layer. If the coder determines in block 405 that the reference layer codec is the particular type of codec, the coder disables inter-layer motion prediction in block 415. In one example, disabling inter-layer motion prediction may comprise not allowing the motion information of the reference layer to be used for coding the current layer. In another embodiment, disabling inter-layer motion prediction may comprise coding the current layer without using the motion information of the reference layer. The method 400 ends at block 420.

[00156] As discussed above, one or more components of video encoder 20 of **FIG. 2A**, video encoder 21 of **FIG. 2B**, video decoder 30 of **FIG. 3A**, or video decoder 31 of **FIG. 3B** (e.g., inter-layer prediction unit 128 and/or inter-layer prediction unit 166) may be used to implement any of the techniques discussed in the present disclosure, such as determining whether the reference layer codec is the particular type of codec, enabling inter-layer motion prediction, and disabling inter-layer motion prediction.

Example Flowchart #2

[00157] **FIG. 5** is a flowchart illustrating a method 500 for coding video information, according to an embodiment of the present disclosure. The steps illustrated in **FIG. 5** may be performed by an encoder (e.g., the video encoder as shown in **FIG. 2A** or **FIG. 2B**), a decoder (e.g., the video decoder as shown in **FIG. 3A** or **FIG. 3B**), or any other component. For convenience, method 500 is described as performed by a coder, which may be the encoder, the decoder, or another component.

[00158] The method 500 begins at block 501. In block 505, the coder determines whether a reference layer codec is a particular type of codec. The way the coder makes the determination in block 505 may be similar to that employed by the coder in block 405 of **FIG. 4**. If the coder determines that the reference layer codec is not the particular type of codec, the method 500 ends at block 520. On the other hand, if the coder determines that the reference layer codec is the particular type of codec, the coder processes an indication that the motion information of the reference layer cannot be used to code the enhancement layer in block 510. In one embodiment, processing the indication that the motion information of the reference layer cannot be used to code the enhancement layer comprises signaling or receiving one or more syntax elements or flags that indicate that motion information of the reference layer cannot be used to code the enhancement layer. In another embodiment, processing the indication that the motion information of the reference layer cannot be used to code the enhancement layer comprises signaling or receiving one or more syntax elements or flags that indicate that information of the reference layer that can be used to code the enhancement layer is restricted to texture information. In another embodiment, processing the indication that the motion information of the reference layer cannot be used to code the enhancement layer comprises refraining from processing a flag or syntax element that indicates whether the reference layer is a direct reference layer of the enhancement layer. In another embodiment, processing the indication that the motion information of the reference layer cannot be used to code the enhancement layer comprises refraining from processing a flag or syntax element that indicates a dependency type between the reference layer and the enhancement layer. In another embodiment, the indication of the prediction type is not signaled in the bitstream, and the prediction type is derived from (or determined based on) available or existing flags, syntax elements, or information. For example, the coder may determine that only texture information of the base layer is to be used for inter-layer prediction if the base layer is coded with non-HEVC codec scheme (e.g., AVC). In block 515, the coder (e.g., in response to processing the indication that the motion information of the reference layer cannot be used to code the enhancement layer) codes the enhancement layer without using the motion information of the reference layer. The method 500 ends at block 520.

[00159] As discussed above, one or more components of video encoder 20 of **FIG. 2A**, video encoder 21 of **FIG. 2B**, video decoder 30 of **FIG. 3A**, or video decoder 31 of **FIG. 3B** (e.g., inter-layer prediction unit 128 and/or inter-layer prediction unit 166) may be used to implement any of the techniques discussed in the present disclosure, such as determining whether the reference layer codec is the particular type of codec, processing an indication that the motion information of the reference layer cannot be used to code the enhancement layer, and coding the enhancement layer without using the motion information of the reference layer.

[00160] In the method 500, one or more of the blocks shown in **FIG. 5** may be removed (e.g., not performed) and/or the order in which the method is performed may be switched. For example, although block 515 is shown in **FIG. 5**, actually coding the enhancement layer need not be part of the method 500 and thus omitted from the method 500. Thus, the embodiments of the present disclosure are not limited to or by the example shown in **FIG. 5**, and other variations may be implemented without departing from the spirit of this disclosure.

Other Considerations

[00161] Information and signals disclosed herein may be represented using any of a variety of different technologies and techniques. For example, data, instructions, commands, information, signals, bits, symbols, and chips that may be referenced throughout the above description may be represented by voltages, currents, electromagnetic waves, magnetic fields or particles, optical fields or particles, or any combination thereof.

[00162] The various illustrative logical blocks, modules, circuits, and algorithm steps described in connection with the embodiments disclosed herein may be implemented as electronic hardware, computer software, or combinations of both. To clearly illustrate this interchangeability of hardware and software, various illustrative components, blocks, modules, circuits, and steps have been described above generally in terms of their functionality. Whether such functionality is implemented as hardware or software depends upon the particular application and design constraints imposed on the overall system. Skilled artisans may implement the described functionality in varying ways for each particular application, but such implementation decisions should not be interpreted as causing a departure from the scope of the present invention.

[00163] The techniques described herein may be implemented in hardware, software, firmware, or any combination thereof. Such techniques may be implemented in any of a variety of devices such as general purposes computers, wireless communication device handsets, or integrated circuit devices having multiple uses including application in wireless communication device handsets and other devices. Any features described as modules or components may be implemented together in an integrated logic device or separately as discrete but interoperable logic devices. If implemented in software, the techniques may be realized at least in part by a computer-readable data storage medium comprising program code including instructions that, when executed, performs one or more of the methods described above. The computer-readable data storage medium may form part of a computer program product, which may include packaging materials. The computer-readable medium may comprise memory or data storage media, such as random access memory (RAM) such as synchronous dynamic random access memory (SDRAM), read-only memory (ROM), non-volatile random access memory (NVRAM), electrically erasable programmable read-only memory (EEPROM), FLASH memory, magnetic or optical data storage media, and the like. The techniques additionally, or alternatively, may be realized at least in part by a computer-readable communication medium that carries or communicates program code in the form of instructions or data structures and that can be accessed, read, and/or executed by a computer, such as propagated signals or waves.

[00164] The program code may be executed by a processor, which may include one or more processors, such as one or more digital signal processors (DSPs), general purpose microprocessors, an application specific integrated circuits (ASICs), field programmable logic arrays (FPGAs), or other equivalent integrated or discrete logic circuitry. Such a processor may be configured to perform any of the techniques described in this disclosure. A general purpose processor may be a microprocessor; but in the alternative, the processor may be any conventional processor, controller, microcontroller, or state machine. A processor may also be implemented as a combination of computing devices, e.g., a combination of a DSP and a microprocessor, a plurality of microprocessors, one or more microprocessors in conjunction with a DSP core, or any other such configuration. Accordingly, the term "processor," as used herein may refer to any of the foregoing structure, any combination of the foregoing structure, or any other structure or apparatus suitable for implementation of the techniques described herein. In addition, in some aspects, the functionality described herein may be provided within dedicated software modules or hardware modules configured for encoding and decoding, or incorporated in a combined video encoder-decoder (CODEC). Also, the techniques could be fully implemented in one or more circuits or logic elements.

[00165] 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 inter-operative hardware units, including one or more processors as described above, in conjunction with suitable software and/or firmware.

[00166] Various embodiments of the invention have been described. These and other embodiments are within the scope of the following claims.

CLAIMS:

1. An apparatus configured to code video information, the apparatus comprising:

a memory configured to store video data associated with a reference layer, an enhancement layer, and one or more additional layers, the reference layer associated with a reference layer (RL) codec and the enhancement layer associated an enhancement layer (EL) codec, the reference layer comprising a first information portion and a second information portion, wherein the first information portion comprises texture information and the second information portion comprises motion information; and

a processor in communication with the memory, the processor configured to:

determine whether the RL codec associated with the reference layer is a first type of codec;

based on a determination that the RL codec is the first type of codec, signal or receive, for each respective layer of the one or more additional layers, one or more flags or syntax elements indicative of whether information of the respective layer can be used to code the enhancement layer;

based on the determination that the RL codec is the first type of codec, refrain from signaling or receiving one or more flags or syntax elements indicative of whether information of the reference layer can be used to code the enhancement layer and infer that a flag or syntax element indicative of whether the information of the reference layer can be used to code the enhancement layer has a first value without signaling or receiving the flag or syntax element; and

based on the determination that the RL codec is the first type of codec, code the enhancement layer using only the first information portion and not using the second information portion to code the enhancement layer.

2. The apparatus of claim 1, wherein the processor is further configured to determine whether the RL codec associated with the reference layer is different from the EL codec associated with the enhancement layer.

Date Reçue/Date Received 2020-10-02

3. The apparatus of claim 1, wherein the processor is further configured to determine a value of a flag or syntax element indicative of whether the RL codec is the first type of codec.

4. The apparatus of claim 1, wherein the processor is further configured to refrain from signaling a flag or syntax element indicative of whether motion information of the reference layer can be used to code the enhancement layer and to determine that motion information of the reference layer is not to be used to code the enhancement layer based on whether the RL codec is a codec other than a High Efficiency Video Coding (HEVC) codec.

5. The apparatus of claim 1, wherein the processor is further configured to refrain from signaling a flag or syntax element indicative of whether motion information of the reference layer can be used to code the enhancement layer and to determine that motion information of the reference layer is not to be used to code the enhancement layer based on whether the RL codec is an Advanced Video Coding (AVC) codec.

6. The apparatus of claim 1, wherein the processor is further configured to signal or receive, in response to a determination that the RL codec is not the first type of codec, a flag or syntax element indicative of whether (i) motion information, (ii) texture information, (iii) both motion information and texture information, or (iv) neither motion information nor texture information of the reference layer can be used to code the enhancement layer.

7. The apparatus of claim 1, wherein the processor is further configured to determine, without signaling or receiving any additional flag or syntax element indicative of whether motion information of the reference layer can be used to code the enhancement layer, that motion information of the reference layer is not to be used to code the enhancement layer based on whether the RL codec is a codec other than a High Efficiency Video Coding (HEVC) codec.

8. The apparatus of claim 1, wherein the processor is further configured to signal or receive one or more syntax elements or flags indicative of whether information of the reference layer that can be used to code the enhancement layer is restricted to texture information.

Date Reçue/Date Received 2020-10-02

9. The apparatus of claim 1, wherein the processor is further configured to refrain from processing a flag or syntax element indicative of whether the reference layer is a direct reference layer of the enhancement layer.

10. The apparatus of claim 1, wherein the processor is further configured to refrain from processing a flag or syntax element indicative of a dependency type between the reference layer and the enhancement layer.

11. The apparatus of claim 1, wherein the first type of codec comprises a codec other than a High Efficiency Video Coding (HEVC) codec.

12. The apparatus of claim 1, wherein the first type of codec comprises an Advanced Video Coding (AVC) codec.

13. The apparatus of claim 1, wherein the RL codec comprises an AVC codec, and the EL codec comprises a High Efficiency Video Coding (HEVC) codec.

14. The apparatus of claim 1, wherein the apparatus comprises an encoder, and wherein the processor is further configured to encode the video data in a video bitstream.

15. The apparatus of claim 1, wherein the apparatus comprises a decoder, and wherein the processor is further configured to decode the video data in a video bitstream.

16. The apparatus of claim 1, wherein the apparatus comprises a device selected from a group consisting one or more of: computers, notebooks, laptops, computers, tablet computers, set-top boxes, telephone handsets, smart phones, smart pads, televisions, cameras, display devices, digital media players, video gaming consoles, and in-car computers.

17. A method of coding video information, the method comprising:

determining whether a reference layer (RL) codec associated with a reference layer is a first type of codec, the reference layer comprising a first information portion and a second information portion, wherein the first information portion comprises texture information and the second information portion comprises motion information;

Date Reçue/Date Received 2020-10-02

based on a determination that the RL codec is the first type of codec, signaling or receiving, for each respective layer of one or more additional layers, one or more flags or syntax elements indicative of whether information of the respective layer can be used to code an enhancement layer;

based on the determination that the RL codec is the first type of codec, refraining from signaling or receiving one or more flags or syntax elements indicative of whether information of the reference layer can be used to code the enhancement layer and inferring that a flag or syntax element indicative of whether the information of the reference layer can be used to code the enhancement layer has a first value without signaling or receiving the flag or syntax element;

based on the determination that the RL codec is the first type of codec, code the enhancement layer using only the first information portion and not using the second information portion to code the enhancement layer.

18. The method of claim 17, further comprising determining whether the RL codec associated with the reference layer is different from a second codec associated with the enhancement layer.

19. The method of claim 17, further comprising determining a value of a flag or syntax element indicative of whether the RL codec is the first type of codec.

20. The method of claim 17, further comprising refraining from signaling a flag or syntax element indicative of whether motion information of the reference layer can be used to code the enhancement layer, and determining that motion information of the reference layer is not to be used to code the enhancement layer based on whether the RL codec is a codec other than a High Efficiency Video Coding (HEVC) codec.

21. The method of claim 17, further comprising refraining from signaling a flag or syntax element indicative of whether motion information of the reference layer can be used to code the enhancement layer, and determining that motion information of the reference layer is not to be used to code the enhancement layer based on whether the RL codec is an Advanced Video Coding (AVC) codec.

Date Reçue/Date Received 2020-10-02

22. The method of claim 17, further comprising signaling or receiving, in response to a determination that the RL codec is not the first type of codec, a flag or syntax element indicative of whether (i) motion information, (ii) texture information, (iii) both motion information and texture information, or (iv) neither motion information nor texture information of the reference layer can be used to code the enhancement layer.

23. The method of claim 17, further comprising determining, without signaling or receiving any additional flag or syntax element indicative of whether motion information of the reference layer can be used to code the enhancement layer, that motion information of the reference layer is not to be used to code the enhancement layer based on whether the RL codec is a codec other than a High Efficiency Video Coding (HEVC) codec.

24. The method of claim 17, further comprising signaling or receiving one or more syntax elements or flags indicative of whether information of the reference layer that can be used to code the enhancement layer is restricted to texture information.

25. The method of claim 17, further comprising refraining from processing a flag or syntax element indicative of whether the reference layer is a direct reference layer of the enhancement layer.

26. The method of claim 17, further comprising refraining from processing a flag or syntax element indicative of a dependency type between the reference layer and the enhancement layer.

27. Non-transitory physical computer storage having stored thereon computer executable code that, when executed, causes an apparatus to:

store video data associated with a reference layer, an enhancement layer, and one or more additional layers, the reference layer associated with a reference layer (RL) codec and the enhancement layer associated an enhancement layer (EL) codec, the reference layer comprising a first information portion and a second information portion, wherein the first information portion comprises texture information and the second information portion comprises motion information;

Date Reçue/Date Received 2020-10-02

determine whether the RL codec associated with the reference layer is a first type of codec;

based on a determination that the RL codec is the first type of codec, signal or receive, for each respective layer of the one or more additional layers, one or more flags or syntax elements indicative of whether information of the respective layer can be used to code the enhancement layer;

based on the determination that the RL codec is the first type of codec, refrain from signaling or receiving one or more flags or syntax elements indicative of whether information of the reference layer can be used to code the enhancement layer and infer that a flag or syntax element indicative of whether the information of the reference layer can be used to code the enhancement layer has a first value without signaling or receiving the flag or syntax element;

based on the determination that the RL codec is the first type of codec code the enhancement layer using only the first information portion and not using the second information portion to code the enhancement layer.

28. The non-transitory physical computer storage of claim 27, wherein the computer executable code further causes the apparatus to determine, without signaling or receiving any additional flag or syntax element indicative of whether motion information of the reference layer can be used to code the enhancement layer, that motion information of the reference layer is not to be used to code the enhancement layer based on whether the RL codec is a codec other than a High Efficiency Video Coding (HEVC) codec.

29. A video coding device configured to code video information, the video coding device comprising:

means for storing video data associated with a reference layer, an enhancement layer, and one or more additional layers, the reference layer associated with a reference layer (RL) codec and the enhancement layer associated an enhancement layer (EL) codec, the reference layer comprising a first information portion and a second information portion,

Date Reçue/Date Received 2020-10-02

wherein the first information portion comprises texture information and the second information portion comprises motion information;

means for determining whether the RL codec associated with the reference layer is a first type of codec;

means for signaling or receiving, for each respective layer of the one or more additional layers one or more flags or syntax elements indicative of whether information of the respective layer can be used to code the enhancement layer based on a determination that the RL codec is the first type of codec,

wherein the means for signaling or receiving is configured to:

based on the determination that the RL codec is the first type of codec, refrain from signaling or receiving one or more flags or syntax elements indicative of whether information of the reference layer can be used to code the enhancement layer and infer that a flag or syntax element indicative of whether the information of the reference layer can be used to code the enhancement layer has a first value without signaling or receiving the flag or syntax element; and

based on the determination that the RL codec is the first type of codec code the enhancement layer using only the first information portion and not using the second information portion to code the enhancement layer.

30. The video coding device of claim 29, further comprising means for determining, without signaling or receiving any additional flag or syntax element indicative of whether motion information of the reference layer can be used to code the enhancement layer, that motion information of the reference layer is not to be used to code the enhancement layer based on whether the RL codec is a codec other than a High Efficiency Video Coding (HEVC) codec.

31. The apparatus of claim 1, wherein the flag or syntax element having the first value indicates that sample information of the reference layer is allowed to be used to code the

Date Reçue/Date Received 2020-10-02

enhancement layer and that motion information of the reference layer is not allowed to be used to code the enhancement layer.

32. The method of claim 17, wherein the flag or syntax element having the first value indicates that sample information of the reference layer is allowed to be used to code the enhancement layer and that motion information of the reference layer is not allowed to be used to code the enhancement layer.

33. An apparatus configured to code video information, the apparatus comprising:

a memory configured to store video data associated with a reference layer and an enhancement layer, the reference layer associated with a reference layer (RL) codec and the enhancement layer associated an enhancement layer (EL) codec, the reference layer comprising a first information portion and a second information portion, wherein the first information portion comprises texture information and the second information portion comprises motion information; and

a processor in communication with the memory, the processor configured to:

determine whether the RL codec associated with the reference layer is a first type of codec;

based on a determination that the RL codec is the first type of codec, refrain from signaling or receiving one or more flags or syntax elements indicative of whether information of the reference layer can be used to code the enhancement layer,

based on the determination that the RL codec is the first type of codec, infer that a flag or syntax element indicative of whether the information of the reference layer can be used to code the enhancement layer has a first value without signaling or receiving the flag or syntax element, and

based on the determination that the RL codec is the first type of codec, code the enhancement layer using only the first information portion and not using the second information portion to code the enhancement layer.

Date Reçue/Date Received 2020-10-02

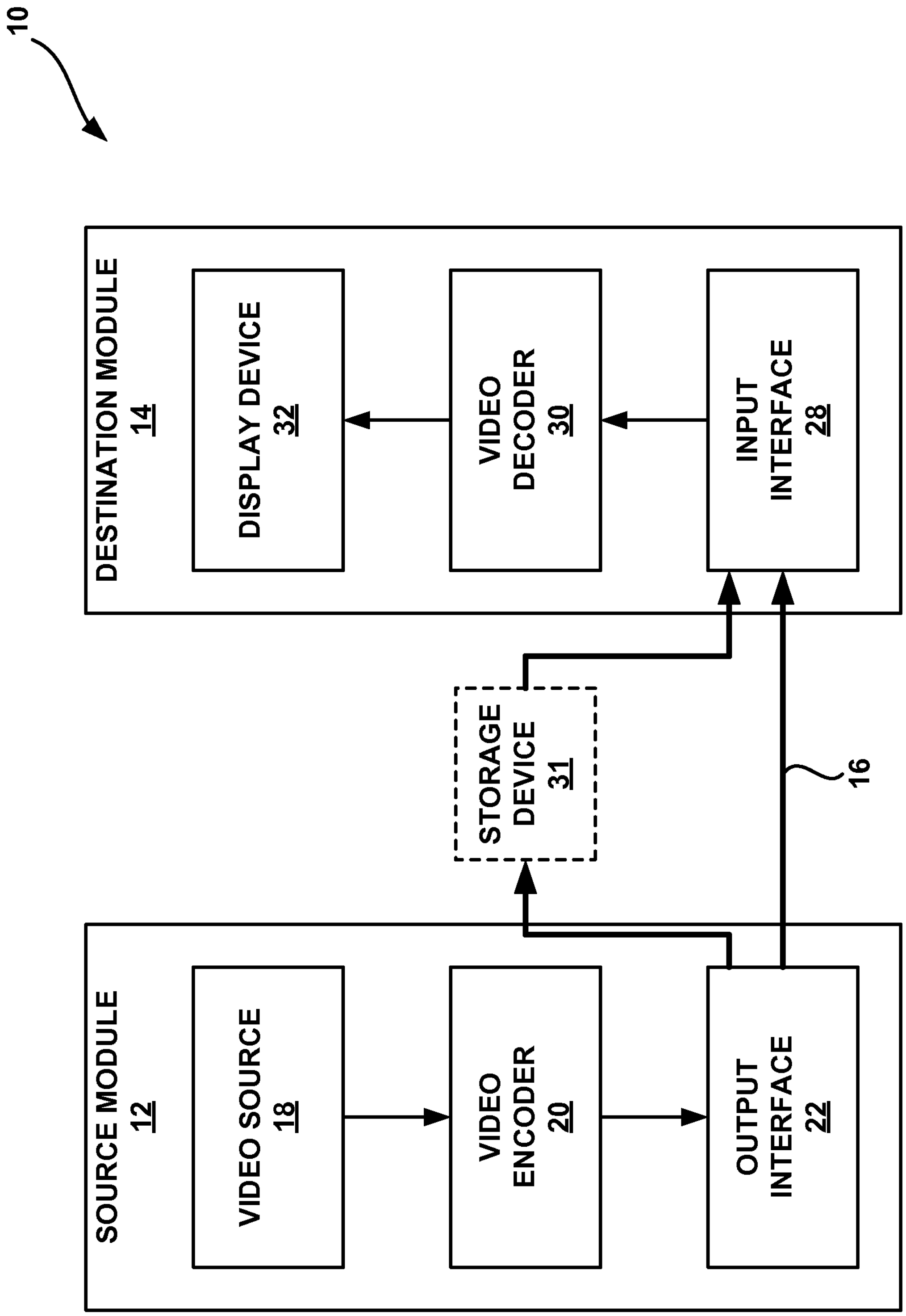

10
DESTINATION MODULE 14
DISPLAY DEVICE 32
VIDEO DECODER 30
INPUT INTERFACE 28
STORAGE DEVICE 31
16
SOURCE MODULE 12
VIDEO SOURCE 18
VIDEO ENCODER 20
OUTPUT INTERFACE 22

FIG. 1A

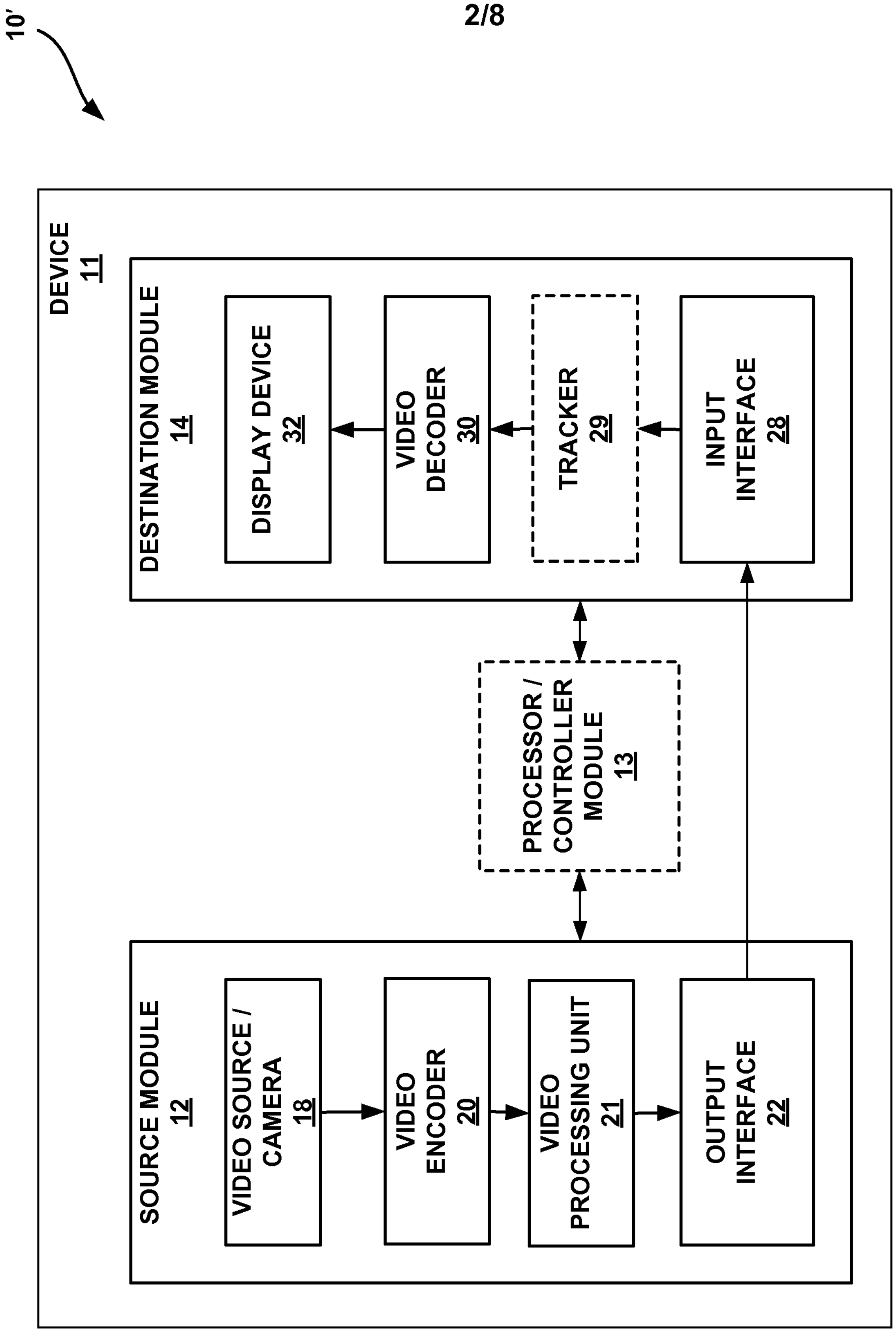

10'
DEVICE 11
DESTINATION MODULE 14
DISPLAY DEVICE 32
VIDEO DECODER 30
TRACKER 29
INPUT INTERFACE 28
PROCESSOR / CONTROLLER MODULE 13
SOURCE MODULE 12
VIDEO SOURCE / CAMERA 18
VIDEO ENCODER 20
VIDEO PROCESSING UNIT 21
OUTPUT INTERFACE 22

FIG. 1B

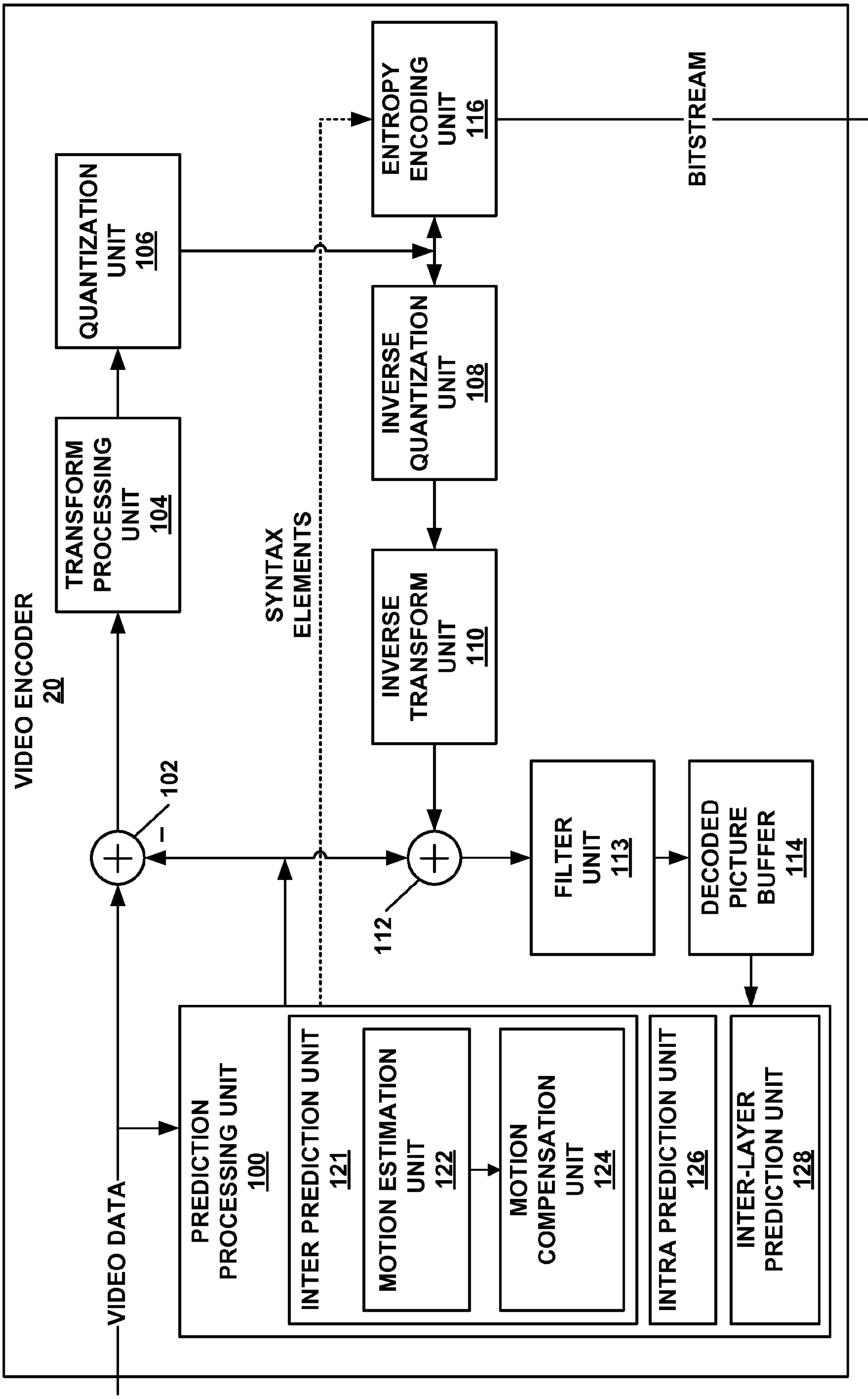

VIDEO ENCODER 20
VIDEO DATA
PREDICTION PROCESSING UNIT 100
INTER PREDICTION UNIT 121
MOTION ESTIMATION UNIT 122
MOTION COMPENSATION UNIT 124
INTRA PREDICTION UNIT 126
INTER-LAYER PREDICTION UNIT 128
102
–
112
TRANSFORM PROCESSING UNIT 104
QUANTIZATION UNIT 106
SYNTAX ELEMENTS
ENTROPY ENCODING UNIT 116
BITSTREAM
INVERSE QUANTIZATION UNIT 108
INVERSE TRANSFORM UNIT 110
FILTER UNIT 113
DECODED PICTURE BUFFER 114

FIG. 2A

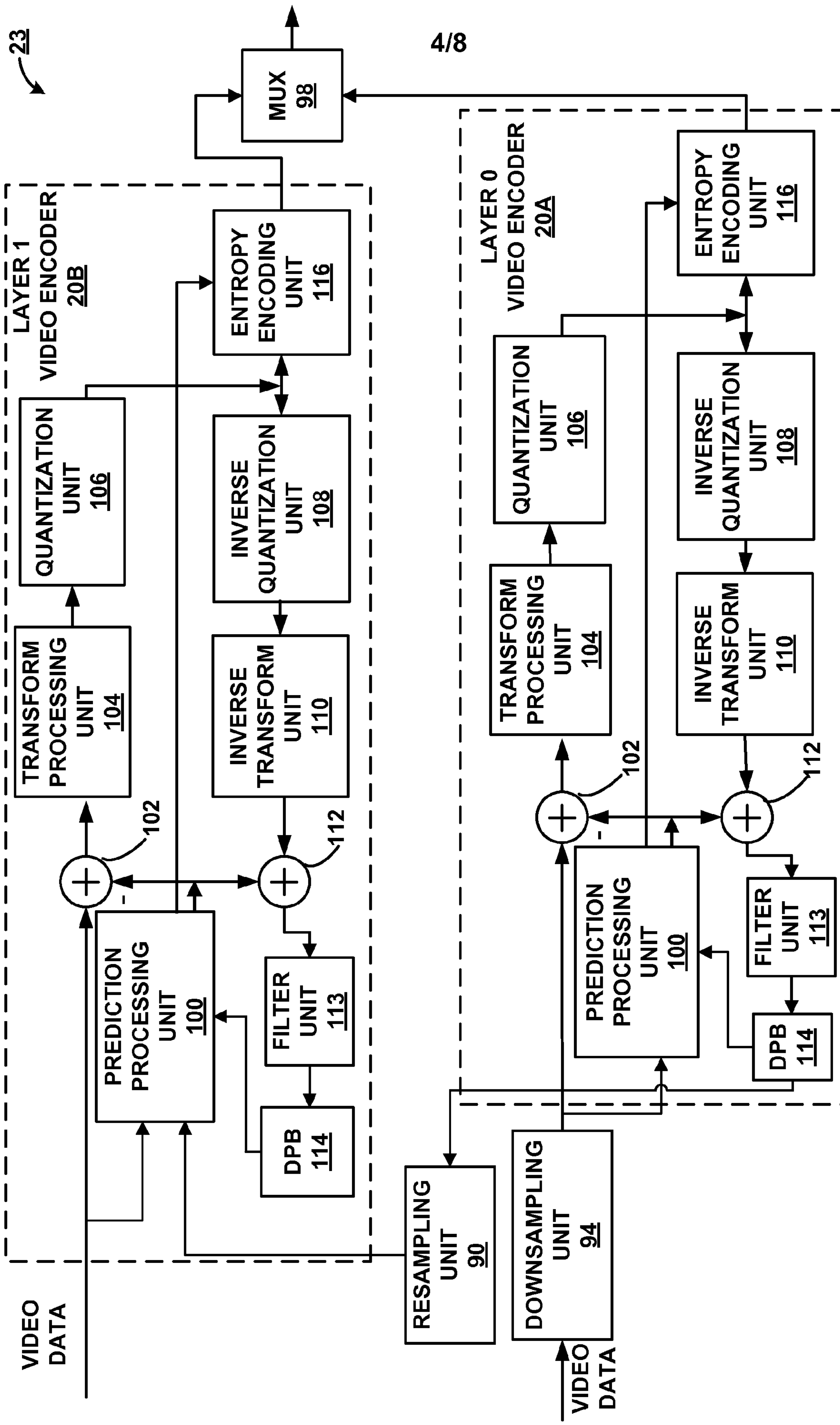

23
LAYER 1
VIDEO ENCODER
20B
VIDEO DATA
PREDICTION PROCESSING UNIT 100
102
TRANSFORM PROCESSING UNIT 104
QUANTIZATION UNIT 106
ENTROPY ENCODING UNIT 116
INVERSE QUANTIZATION UNIT 108
INVERSE TRANSFORM UNIT 110
112
FILTER UNIT 113
DPB 114
MUX 98
RESAMPLING UNIT 90
DOWNSAMPLING UNIT 94
LAYER 0
VIDEO ENCODER
20A
VIDEO DATA
PREDICTION PROCESSING UNIT 100
102
TRANSFORM PROCESSING UNIT 104
QUANTIZATION UNIT 106
ENTROPY ENCODING UNIT 116
INVERSE QUANTIZATION UNIT 108
INVERSE TRANSFORM UNIT 110
112
FILTER UNIT 113
DPB 114

FIG. 2B

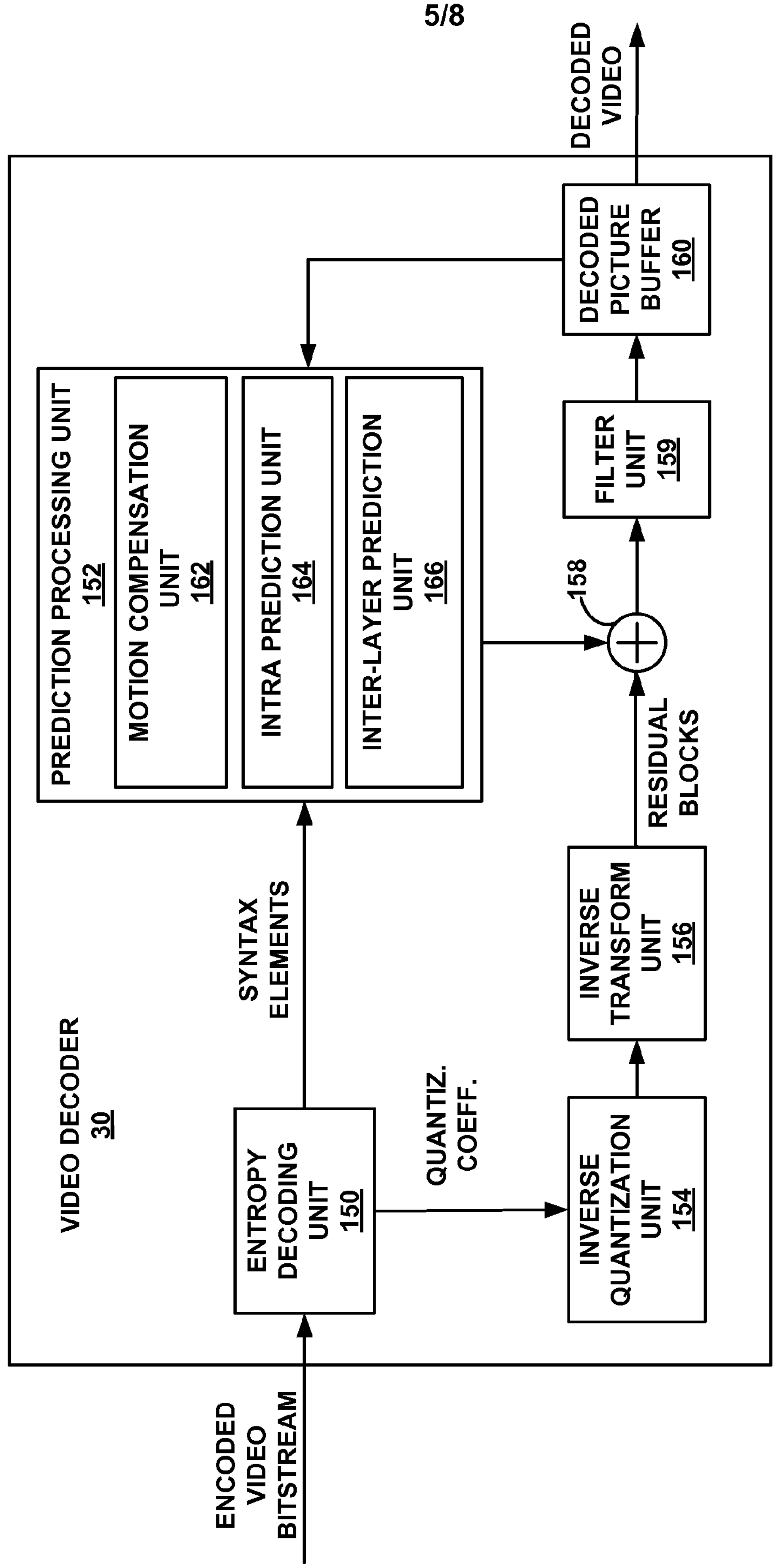

VIDEO DECODER 30
ENCODED VIDEO BITSTREAM
ENTROPY DECODING UNIT 150
SYNTAX ELEMENTS
QUANTIZ. COEFF.
INVERSE QUANTIZATION UNIT 154
INVERSE TRANSFORM UNIT 156
RESIDUAL BLOCKS
PREDICTION PROCESSING UNIT 152
MOTION COMPENSATION UNIT 162
INTRA PREDICTION UNIT 164
INTER-LAYER PREDICTION UNIT 166
158
FILTER UNIT 159
DECODED PICTURE BUFFER 160
DECODED VIDEO

FIG. 3A

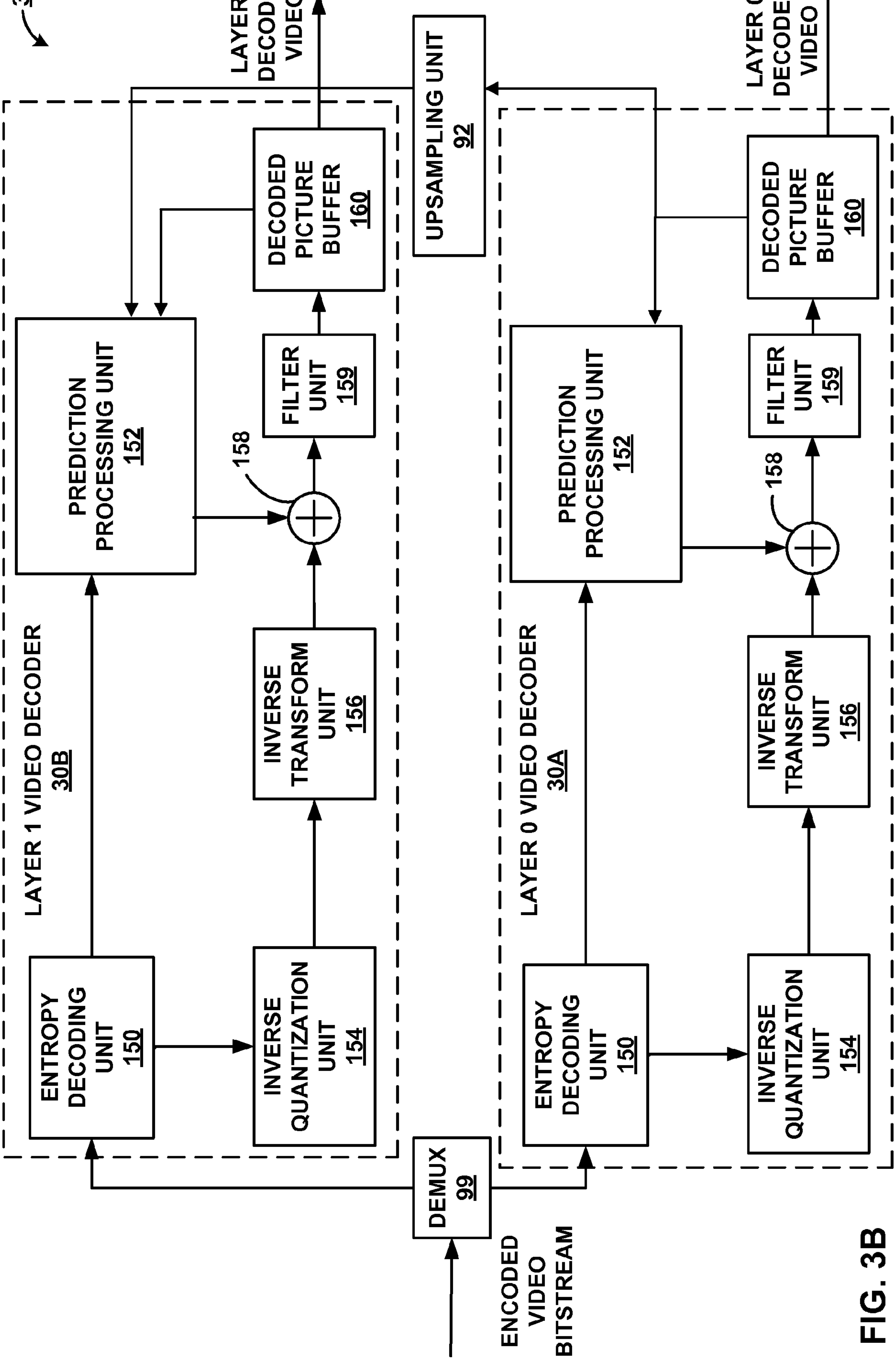

33
LAYER 1 VIDEO DECODER 30B
ENTROPY DECODING UNIT 150
PREDICTION PROCESSING UNIT 152
INVERSE QUANTIZATION UNIT 154
INVERSE TRANSFORM UNIT 156
158
FILTER UNIT 159
DECODED PICTURE BUFFER 160
LAYER 1 DECODED VIDEO
UPSAMPLING UNIT 92
DEMUX 99
ENCODED VIDEO BITSTREAM
LAYER 0 VIDEO DECODER 30A
ENTROPY DECODING UNIT 150
PREDICTION PROCESSING UNIT 152
INVERSE QUANTIZATION UNIT 154
INVERSE TRANSFORM UNIT 156
158
FILTER UNIT 159
DECODED PICTURE BUFFER 160
LAYER 0 DECODED VIDEO

FIG. 3B

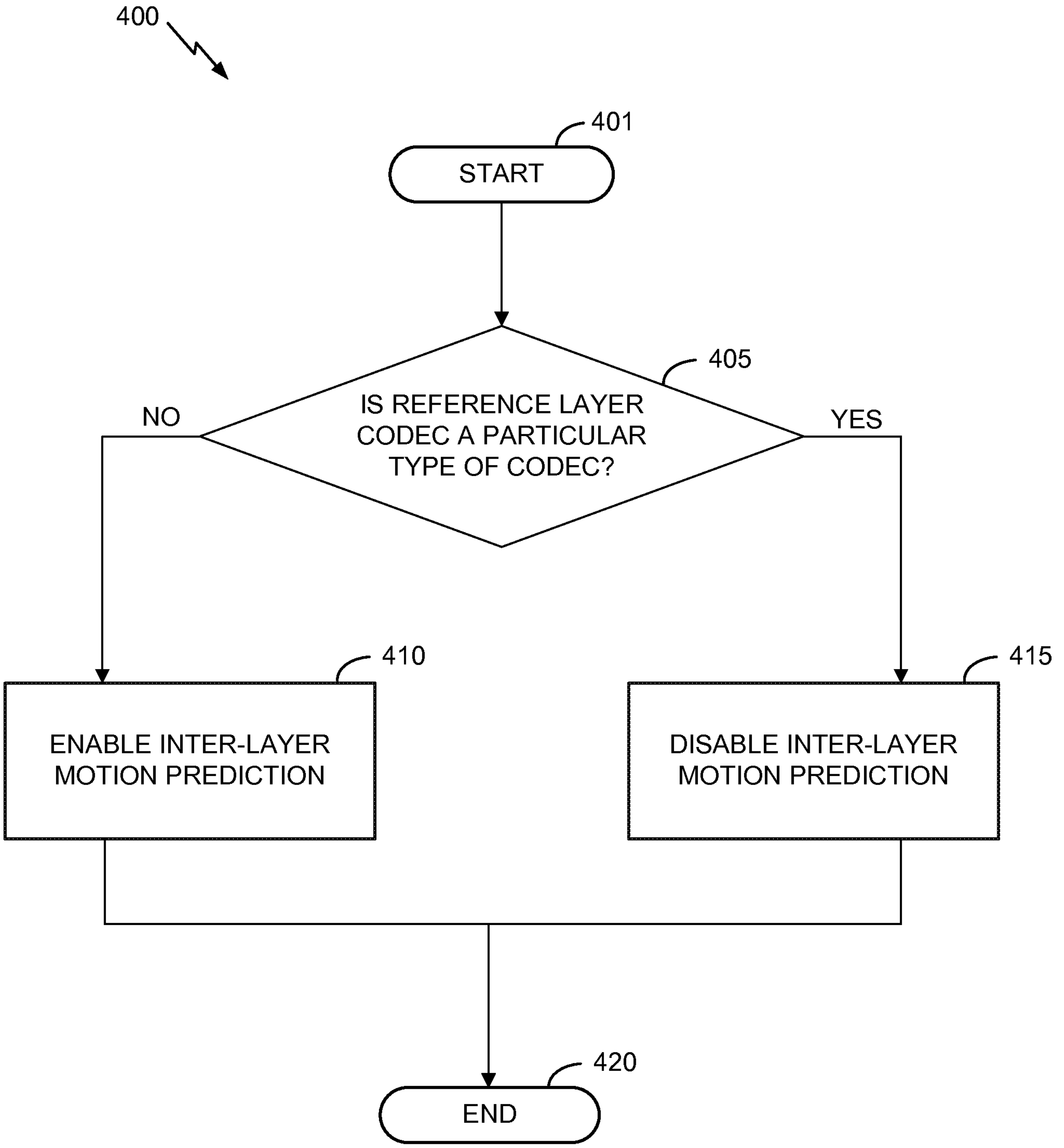

400
401
START
405
IS REFERENCE LAYER CODEC A PARTICULAR TYPE OF CODEC?
NO
YES
410
ENABLE INTER-LAYER MOTION PREDICTION
415
DISABLE INTER-LAYER MOTION PREDICTION
420
END

FIG. 4

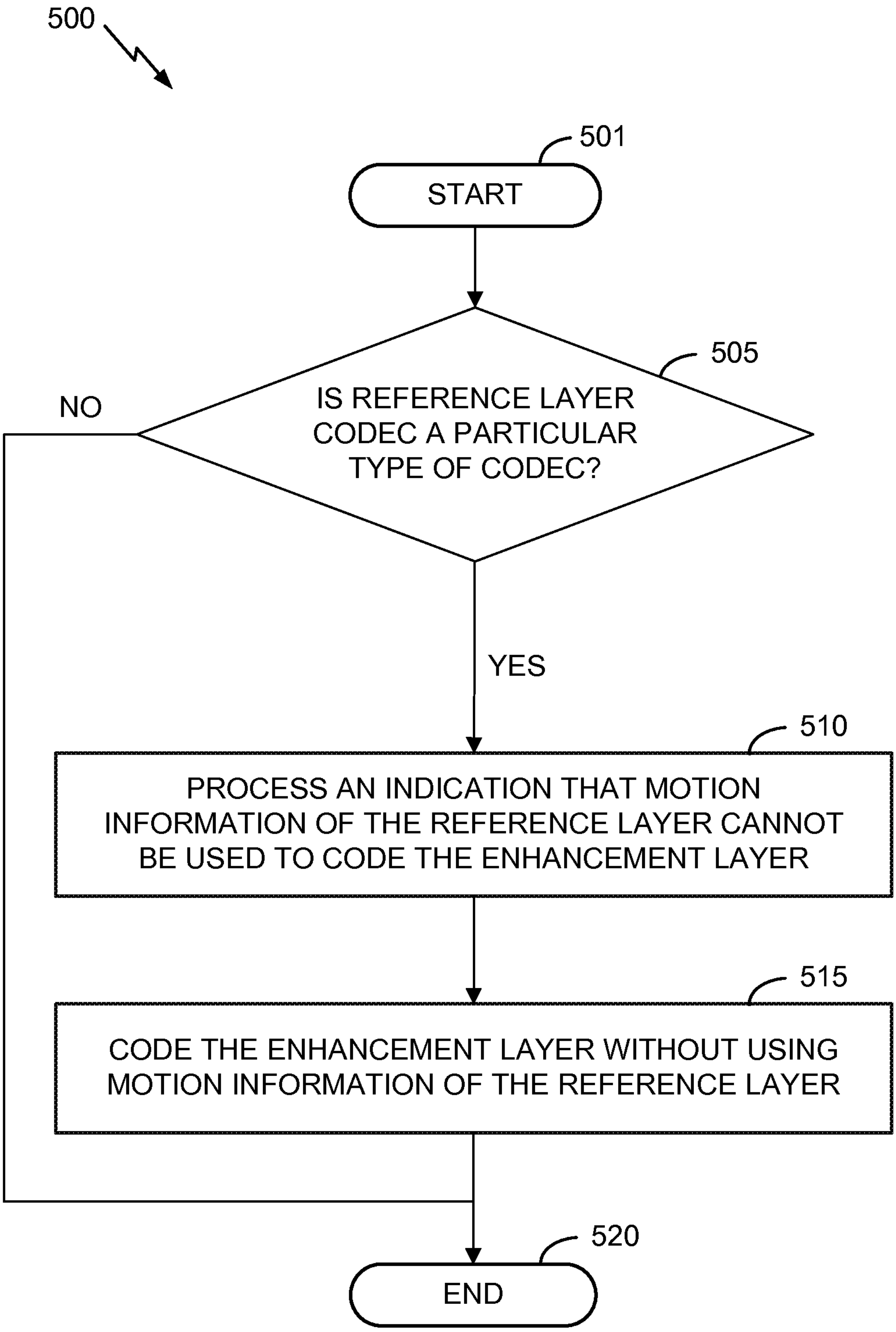
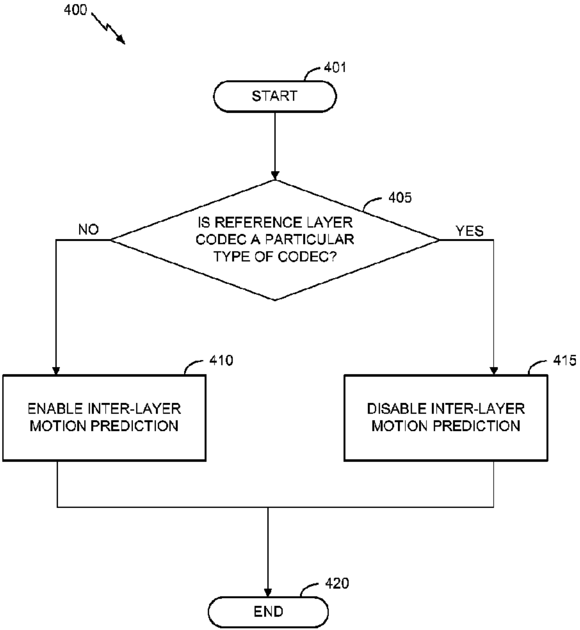

500
501
START
505
IS REFERENCE LAYER CODEC A PARTICULAR TYPE OF CODEC?
NO
YES
510
PROCESS AN INDICATION THAT MOTION INFORMATION OF THE REFERENCE LAYER CANNOT BE USED TO CODE THE ENHANCEMENT LAYER
515
CODE THE ENHANCEMENT LAYER WITHOUT USING MOTION INFORMATION OF THE REFERENCE LAYER
520
END

FIG. 5


400
401
START
405
IS REFERENCE LAYER CODEC A PARTICULAR TYPE OF CODEC?
NO
YES
410
ENABLE INTER-LAYER MOTION PREDICTION
415
DISABLE INTER-LAYER MOTION PREDICTION
420
END