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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
9 July 2015 (09.07.2015)

WIPO | PCT

(10) International Publication Number
WO 2015/103224 A1

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

(21) International Application Number:
PCT/US2014/072698

(22) International Filing Date:
30 December 2014 (30.12.2014)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/923,529 3 January 2014 (03.01.2014) US
14/584,473 29 December 2014 (29.12.2014) US

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

(72) Inventors: **WANG, Ye-Kui**; 5775 Morehouse Drive, San Diego, California 92121-1714 (US). **CHEN, Jianle**; 5775 Morehouse Drive, San Diego, California 92121-1714 (US). **CHEN, Ying**; 5775 Morehouse Drive, San Diego, California 92121-1714 (US). **HENDRY, Fnu**; 5775 Morehouse Drive, San Diego, California 92121-1714 (US).

(74) Agents: **ABUMERI, Mark M.** et al.; 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, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

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

[Continued on next page]

(54) Title: SUPPORT OF BASE LAYER OF A DIFFERENT CODEC IN MULTI-LAYER VIDEO CODING

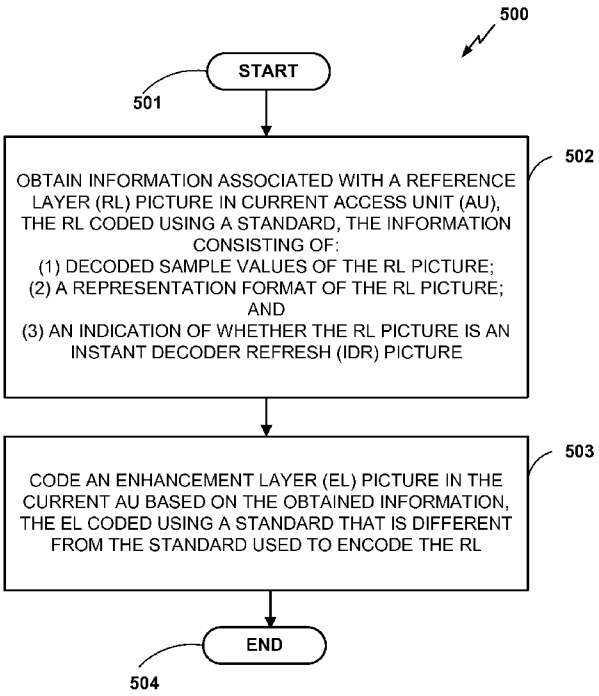

FIG. 5

(57) Abstract: An apparatus for coding video information according to certain aspects includes a memory and a processor. The memory unit is configured to store video information associated with an enhancement layer (EL) and a corresponding reference layer (RL). The processor is configured to: code an EL picture in a current access unit (AU), the EL coded using a first standard that is different from a second standard that is used to code the RL, wherein the coding of the EL picture is based on information associated with a RL picture in the current access unit, the information associated with the RL picture provided by an external means and consisting of: (1) decoded sample values of the RL picture; (2) a representation format of the RL picture; and (3) an indication of whether the RL picture is an instantaneous decoding refresh (IDR) picture.

WO 2015/103224 A1

Published:

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

SUPPORT OF BASE LAYER OF A DIFFERENT CODEC IN MULTI-LAYER VIDEO CODING

TECHNICAL FIELD

[0001] This disclosure relates to the field of video coding and compression, particularly to multi-layer video coding that includes 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 middle layer (e.g., a layer that is neither the lowest layer nor the highest layer) may be an EL for the layers below the middle layer, 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 the middle layer. 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] An apparatus for coding video information according to certain aspects includes a memory and a processor. The memory unit is configured to store video information associated with an enhancement layer (EL) and a corresponding reference layer (RL). The processor is configured to code an EL picture in a current access unit (AU), the EL coded using a first standard that is different from a second standard that is used to code the RL, wherein the coding of the EL picture is based on information associated with a RL picture in the current access unit, the information associated with the RL picture provided by an external means and consisting of: (1) decoded sample values of the RL picture; (2) a representation format of the RL picture; and (3) an indication of whether the RL picture is an instantaneous decoding refresh (IDR) picture.

[0007] 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. The details of one or more examples are set forth in the accompanying drawings and the description below, which are not intended to limit the full scope of the inventive concepts described herein. Other features, objects, and advantages will be apparent from the description and drawings, and from the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] 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.

[0009] 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.

[0010] 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.

[0011] 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.

[0012] 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.

[0013] 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.

[0014] FIG. 4 is a block diagram illustrating an example configuration of pictures in different layers, according to one embodiment of the present disclosure.

[0015] FIG. 5 is a flowchart illustrating a method of coding video information, according to one embodiment of the present disclosure.

DETAILED DESCRIPTION

[0016] In general, this disclosure relates 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 extension of HEVC, which may be referred to as SHVC.

[0017] In the description below, H.264/Advanced Video Coding (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: International Telecommunication Union (ITU) Telecommunication Standardization Sector (ITU-T) H.261, International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) (ISO/IEC) Moving Picture Experts Group (MPEG) 1 (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.

[0018] HEVC generally follows the framework of previous video coding standards in many respects. The unit of prediction in HEVC is different from the units of prediction (e.g., macroblock) in certain previous video coding standards. In fact, the concept of a macroblock does not exist in HEVC as understood in certain previous video coding standards. A 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 HEVC does not restrict the maximum size of CUs 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 a single PU may contain multiple arbitrary shape partitions to effectively code irregular image patterns. TU may be considered the basic unit of transform. TU can be defined independently from the PU; however, the size of a TU may be limited to the size of the CU to which the TU belongs. This separation of the block structure into three different concepts may allow each unit to be optimized according to the respective role of the unit, which may result in improved coding efficiency.

[0019] 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) of video data. A "layer" of video data may generally refer to a sequence of pictures having at least one common characteristic, such as a view, a frame rate, a resolution, or the like. For example, a layer may include video data

associated with a particular view (e.g., perspective) of multi-view video data. As another example, a layer may include video data associated with a particular layer of scalable video data. Thus, this disclosure may interchangeably refer to a layer and a view of video data. That is, a view of video data may be referred to as a layer of video data, and a layer of video data may be referred to as a view of video data. In addition, a multi-layer codec (also referred to as a multi-layer video coder or multi-layer encoder-decoder) may jointly refer to a multiview codec or a scalable codec (e.g., a codec configured to encode and/or decode video data using MV-HEVC, 3D-HEVC, SHVC, or another multi-layer coding technique). Video encoding and video decoding may both generally be referred to as video coding. 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

[0020] 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 sheer quantity of information to be conveyed from an image encoder to an image decoder would render 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.

[0021] 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 SVC and MVC extensions.

[0022] In addition, a new video coding standard, namely 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.

Overview

[0023] In some cases of multi-layer coding, the reference layer (RL) may be coded using one standard and the enhancement layer (EL) may be coded using another standard. For example, the RL may be coded according to H.264/AVC, and the EL may be coded according to H.265/HEVC. Providing support for a RL that is coded using an earlier standard (or coding scheme) or an earlier version of a standard (or coding scheme) can be beneficial since video data coded using an earlier standard or an earlier version of a given standard may be utilized in multi-layer video coding using the current standard or a current version of a standard having an earlier version. However, managing many aspects of processing the RL coded in a different standard or a different version of a standard can cause support for multi-layer coding to become complicated. For example, the encoder or decoder may need to handle output of RL pictures, maintain decoded picture stores for RL pictures, etc.

[0024] In order to address these and other challenges, the techniques according to certain aspects can support multi-layer video coding using different standards or using different versions of a standard in a simplified manner. For instance, the HEVC decoder may minimize managing and processing of RL pictures coded using a standard other than HEVC. For illustrative purposes, the RL will be explained as being coded using H.264/AVC, and the EL will be will be explained as being coded using H.265/HEVC. However, any combination of different standards can be used to code the RL and EL. According to certain aspects, the techniques minimize managing and processing of RL pictures as follows: (1) the decoded RL pictures are provided by external means, and (2) the output of RL pictures, including the synchronization with output of EL pictures, are controlled by external means. External means may refer to a coder (e.g., an encoder or a decoder) that supports the standard used to code the RL. The HEVC decoder can implement certain rules in order to support multi-layer video coding using different standards. The rules and details relating to the rules are explained further below.

Video Coding System

[0025] 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.

[0026] 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.

[0027] 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.

[0028] **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. In addition to video encoders and video decoders, the aspects described in the present application may be extended to other related devices such as transcoders (e.g., devices that can decode a bitstream and re-encode another bitstream) and middleboxes (e.g., devices that can modify, transform, and/or otherwise manipulate a bitstream).

[0029] As shown in **FIG. 1A**, video coding system 10 includes a source device 12 that generates encoded video data to be decoded at a later time by a destination device 14. In the example of **FIG. 1A**, the source device 12 and destination device 14 constitute separate devices. It is noted, however, that the source and destination devices 12, 14 may be on or part of the same device, as shown in the example of **FIG. 1B**.

[0030] With reference once again, to **FIG. 1A**, the source device 12 and the destination device 14 may respectively 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 device 12 and the destination device 14 may be equipped for wireless communication.

[0031] The destination device 14 may receive, via link 16, the encoded video data to be decoded. The link 16 may comprise any type of medium or device capable of moving the encoded video data from the source device 12 to the destination device 14. In the example of **FIG. 1A**, the link 16 may comprise a communication medium to enable the source device 12 to transmit encoded video data to the destination device 14 in real-time. The encoded video data may be modulated according to a communication standard, such as a wireless communication protocol, and transmitted to the destination device 14. The communication medium may comprise any wireless or wired communication medium, such as a radio frequency (RF) spectrum or one or more physical transmission lines. The communication medium may form part of a packet-based network, such as a local area network, a wide-area network, or a global network such as the Internet. The communication medium may include routers, switches, base stations, or any other equipment that may be useful to facilitate communication from the source device 12 to the destination device 14.

[0032] 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, for example, of the destination device 14. 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 device 12. The destination device 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 device 14. Example file servers include a web server (e.g., for a website), a File Transfer Protocol (FTP) server, network attached storage (NAS) devices, or a local disk drive. The destination device 14 may access the encoded video data through any standard data connection, including an Internet connection. This may include a wireless channel (e.g., a wireless local area network (WLAN) connection), a wired connection (e.g., a digital subscriber line (DSL), a 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.

[0033] 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 Hypertext Transfer Protocol (HTTP), 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.

[0034] In the example of **FIG. 1A**, the source device 12 includes a video source 18, a video encoder 20 and the output interface 22. In some cases, the output interface 22 may include a modulator/demodulator (modem) and/or a transmitter. In the source device 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 device 12 and the destination device 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.

[0035] The captured, pre-captured, or computer-generated video may be encoded by the video encoder 20. The encoded video data may be transmitted to the destination device 14 via the output interface 22 of the source device 12. The encoded video data may also (or alternatively) be stored onto the storage device 31 for later access by the destination device 14 or other devices, for decoding and/or playback. The video encoder 20 illustrated in **FIG. 1A** and **1B** may comprise the video encoder 20 illustrated **FIG. 2A**, the video encoder 23 illustrated in **FIG. 2B**, or any other video encoder described herein.

[0036] In the example of **FIG. 1A**, the destination device 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 device 14 may receive the encoded video data over the link 16 and/or from the storage device 31. 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. The video decoder 30 illustrated in **FIG. 1A** and **1B** may comprise the video decoder 30 illustrated **FIG. 3A**, the video decoder 33 illustrated in **FIG. 3B**, or any other video decoder described herein.

[0037] The display device 32 may be integrated with, or external to, the destination device 14. In some examples, the destination device 14 may include an integrated display device and also be configured to interface with an external display device. In other examples, the destination device 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.

[0038] In related aspects, **FIG. 1B** shows an example video encoding and decoding system 10′ wherein the source and destination devices 12, 14 are on or part of a 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 device 13 in operative communication with the source and destination devices 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 device 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.

[0039] The video encoder 20 and the video decoder 30 may operate according to a video compression standard, such as the HEVC, and may conform to a HEVC Test Model (HM). Alternatively, the 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, 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.

[0040] Although not shown in the examples of **FIGS. 1A** and **1B**, the video encoder 20 and the 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).

[0041] 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

[0042] As mentioned briefly above, the 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 the video encoder 20 encodes the video data, the 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.

[0043] To generate the bitstream, the video encoder 20 may perform encoding operations on each picture in the video data. When the video encoder 20 performs encoding operations on the pictures, the 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 (SPSs), picture parameter sets (PPSs), adaptation parameter sets (APSs), and other syntax structures. A SPS may contain parameters applicable to zero or more sequences of pictures. A PPS may contain parameters applicable to zero or more pictures. An 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.

[0044] To generate a coded picture, the 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). The video encoder 20 may use quadtree partitioning to partition the video blocks of treeblocks into video blocks associated with CUs, hence the name "treeblocks."

[0045] In some examples, the 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.

[0046] As part of performing an encoding operation on a picture, the video encoder 20 may perform encoding operations on each slice of the picture. When the video encoder 20 performs an encoding operation on a slice, the 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."

[0047] To generate a coded slice, the video encoder 20 may perform encoding operations on each treeblock in a slice. When the video encoder 20 performs an encoding operation on a treeblock, the video encoder 20 may generate a coded treeblock. The coded treeblock may comprise data representing an encoded version of the treeblock.

[0048] When the video encoder 20 generates a coded slice, the video encoder 20 may perform encoding operations on (e.g., encode) the treeblocks in the slice according to a raster scan order. For example, the 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 the video encoder 20 has encoded each of the treeblocks in the slice.

[0049] 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, the 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, the 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.

[0050] To generate a coded treeblock, the 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, the 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.

[0051] One or more syntax elements in the bitstream may indicate a maximum number of times the 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.

[0052] The 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, the 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 the video encoder 20 performs an encoding operation on a partitioned CU, the 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, the 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.

[0053] 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, the 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, the video encoder 20 may be unable to access information generated by encoding other CUs that neighbor the given CU when encoding the given CU.

[0054] When the video encoder 20 encodes a non-partitioned CU, the 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. The 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. The video encoder 20 may use intra prediction or inter prediction to generate the predicted video block for a PU.

[0055] When the video encoder 20 uses intra prediction to generate the predicted video block of a PU, the video encoder 20 may generate the predicted video block of the PU based on decoded samples of the picture associated with the PU. If the 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 the video encoder 20 uses inter prediction to generate the predicted video block of the PU, the 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 the video encoder 20 uses inter prediction to generate predicted video blocks of the PUs of a CU, the CU is an inter-predicted CU.

[0056] Furthermore, when the video encoder 20 uses inter prediction to generate a predicted video block for a PU, the 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. The video encoder 20 may generate the predicted video block for the PU based on the reference blocks of the PU.

[0057] After the video encoder 20 generates predicted video blocks for one or more PUs of a CU, the 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.

[0058] Furthermore, as part of performing an encoding operation on a non-partitioned CU, the 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.

[0059] The 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.

[0060] After generating a transform coefficient block, the 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.

[0061] The video encoder 20 may associate each CU with a quantization parameter (QP) value. The QP value associated with a CU may determine how the video encoder 20 quantizes transform coefficient blocks associated with the CU. The 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.

[0062] After the video encoder 20 quantizes a transform coefficient block, the video encoder 20 may generate sets of syntax elements that represent the transform coefficients in the quantized transform coefficient block. The 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.

[0063] The bitstream generated by the 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.

[0064] The video decoder 30 may receive the bitstream generated by the video encoder 20. The bitstream may include a coded representation of the video data encoded by the video encoder 20. When the video decoder 30 receives the bitstream, the video decoder 30 may perform a parsing operation on the bitstream. When the video decoder 30 performs the parsing operation, the video decoder 30 may extract syntax elements from the bitstream. The 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 the video encoder 20 to generate the syntax elements.

[0065] After the video decoder 30 extracts the syntax elements associated with a CU, the video decoder 30 may generate predicted video blocks for the PUs of the CU based on the syntax elements. In addition, the video decoder 30 may inverse quantize transform coefficient blocks associated with TUs of the CU. The 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, the 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, the video decoder 30 may reconstruct the video blocks of CUs based on the syntax elements in the bitstream.

Video Encoder

[0066] **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, including but not limited to the methods of support for multi-layer coding using different standards and related processes described in greater detail above and below with respect to FIGS. 4 and 5. 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.

[0067] For purposes of explanation, this disclosure describes the 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.

[0068] The 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.

[0069] In the example of **FIG. 2A**, the video encoder 20 includes a plurality of functional components. The functional components of the 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, the 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.

[0070] The video encoder 20 may receive video data. The video encoder 20 may receive the video data from various sources. For example, the 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, the video encoder 20 may perform an encoding operation on each of the pictures. As part of performing the encoding operation on a picture, the video encoder 20 may perform encoding operations on each slice of the picture. As part of performing an encoding operation on a slice, the video encoder 20 may perform encoding operations on treeblocks in the slice.

[0071] 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.

[0072] 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.

[0073] 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.

[0074] 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.

[0075] The video encoder 20 may perform encoding operations on each non-partitioned CU of a treeblock. When the video encoder 20 performs an encoding operation on a non-partitioned CU, the video encoder 20 generates data representing an encoded representation of the non-partitioned CU.

[0076] 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. The video encoder 20 and the video decoder 30 may support various PU sizes. Assuming that the size of a particular CU is 2Nx2N, the video encoder 20 and the 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. The video encoder 20 and the 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.

[0077] 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.

[0078] 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.

[0079] 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.

[0080] 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.

[0081] 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.

[0082] 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.

[0083] 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.

[0084] 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 the 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. The 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, the video encoder 20 may be able to signal the motion information of the second PU using fewer bits.

[0085] As further discussed below with reference to **FIG. 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 **FIG. 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 **FIG. 5**, either together or separately.

[0086] 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.

[0087] 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.

[0088] 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.

[0089] 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 may be probable that 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.

[0090] 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.

[0091] 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.

[0092] 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.

[0093] 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.

[0094] 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.

[0095] The video encoder 20 may associate a QP value with a CU in various ways. For example, the video encoder 20 may perform a rate-distortion analysis on a treeblock associated with the CU. In the rate-distortion analysis, the video encoder 20 may generate multiple coded representations of the treeblock by performing an encoding operation multiple times on the treeblock. The video encoder 20 may associate different QP values with the CU when the video encoder 20 generates different encoded representations of the treeblock. The 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.

[0096] 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, the video encoder 20 may reconstruct the video block of the CU.

[0097] 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.

[0098] Entropy encoding unit 116 may receive data from other functional components of the 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, the 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.

[0099] 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

[00100] **FIG. 2B** is a block diagram illustrating an example of a multi-layer video encoder 23 (also simply referred to as 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.

[00101] 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.

[00102] 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.

[00103] 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**.

[00104] 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.

[00105] 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.

[00106] 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 device 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

[00107] **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, including but not limited to the methods of support for multi-layer coding using different standards and related processes described in greater detail above and below with respect to FIGS. 4 and 5. 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.

[00108] For purposes of explanation, this disclosure describes the 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.

[00109] In the example of **FIG. 3A**, the video decoder 30 includes a plurality of functional components. The functional components of the 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, the video decoder 30 may perform a decoding pass generally reciprocal to the encoding pass described with respect to the video encoder 20 of **FIG. 2A**. In other examples, the video decoder 30 may include more, fewer, or different functional components.

[00110] The video decoder 30 may receive a bitstream that comprises encoded video data. The bitstream may include a plurality of syntax elements. When the 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.

[00111] 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.

[00112] 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.

[00113] 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.

[00114] After entropy decoding unit 150 performs a parsing operation on a non-partitioned CU, the video decoder 30 may perform a reconstruction operation on the non-partitioned CU. To perform the reconstruction operation on a non-partitioned CU, the 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, the video decoder 30 may reconstruct a residual video block associated with the CU.

[00115] 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 the 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.

[00116] 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 the 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.

[00117] 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 the 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 the video encoder 20 according to received syntax information and use the interpolation filters to produce the predicted video block.

[00118] As further discussed below with reference to **FIG. 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 **FIG. 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 **FIG. 5**, either together or separately.

[00119] 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.

[00120] 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.

[00121] As discussed above, the 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.

[00122] 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, the 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.

[00123] 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, the 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, the 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

[00124] **FIG. 3B** is a block diagram illustrating an example of a multi-layer video decoder 33 (also simply referred to as 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.

[00125] 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.

[00126] 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

[00127] 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**.

[00128] 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.

[00129] 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.

Intra Random Access Point (IRAP) Pictures

[00130] Some video coding schemes may provide various random access points throughout the bitstream such that the bitstream may be decoded starting from any of those random access points without needing to decode any pictures that precede those random access points in the bitstream. In such video coding schemes, all pictures that follow a random access point in output order (e.g., including those pictures that are in the same access unit as the picture providing the random access point) can be correctly decoded without using any pictures that precede the random access point. For example, even if a portion of the bitstream is lost during transmission or during decoding, a decoder can resume decoding the bitstream starting from the next random access point. Support for random access may facilitate, for example, dynamic streaming services, seek operations, channel switching, etc.

[00131] In some coding schemes, such random access points may be provided by pictures that are referred to as intra random access point (IRAP) pictures. For example, a random access point (e.g., provided by an enhancement layer IRAP picture) in an enhancement layer ("layerA") contained in an access unit ("auA") may provide layer-specific random access such that for each reference layer ("layerB") of layerA (e.g., a reference layer being a layer that is used to predict layerA) having a random access point contained in an access unit ("auB") that is in layerB and precedes auA in decoding order (or a random access point contained in auA), the pictures in layerA that follow auB in output order (including those pictures located in auB), are correctly decodable without needing to decode any pictures in layerA that precede auB.

[00132] IRAP pictures may be coded using intra prediction (e.g., coded without referring to other pictures), and may include, for example, instantaneous decoding refresh (IDR) pictures, clean random access (CRA) pictures, and broken link access (BLA) pictures. When there is an IDR picture in the bitstream, all the pictures that precede the IDR picture in decoding order are not used for prediction by pictures that follow the IDR picture in decoding order. When there is a CRA picture in the bitstream, the pictures that follow the CRA picture may or may not use pictures that precede the CRA picture in decoding order for prediction. Those pictures that follow the CRA picture in decoding order but use pictures that precede the CRA picture in decoding order may be referred to as random access skipped leading (RASL) pictures. Another type of picture that follows an IRAP picture in decoding order and

precedes the IRAP picture in output order is a random access decodable leading (RADL) picture, which may not contain references to any pictures that precede the IRAP picture in decoding order. RASL pictures may be discarded by the decoder if the pictures that precede the CRA picture are not available. A BLA picture indicates to the decoder that pictures that precede the BLA picture may not be available to the decoder (e.g., because two bitstreams are spliced together and the BLA picture is the first picture of the second bitstream in decoding order). An access unit (e.g., a group of pictures consisting of all the coded pictures associated with the same output time across multiple layers) containing a base layer picture (e.g., a picture having a layer ID value of 0) that is an IRAP picture may be referred to as an IRAP access unit.

Cross-Layer Alignment of IRAP Pictures

[00133] In SVC, IRAP pictures may not be required to be aligned (e.g., contained in the same access unit) across different layers. For example, if IRAP pictures were required to be aligned, any access unit containing at least one IRAP picture would only contain IRAP pictures. On the other hand, if IRAP pictures were not required to be aligned, in a single access unit, one picture (e.g., in a first layer) may be an IRAP picture, and another picture (e.g., in a second layer) may be a non-IRAP picture. Having such non-aligned IRAP pictures in a bitstream may provide some advantages. For example, in a two-layer bitstream, if there are more IRAP pictures in the base layer than in the enhancement layer, in broadcast and multicast applications, low tune-in delay and high coding efficiency can be achieved.

[00134] In some video coding schemes, a picture order count (POC) may be used to keep track of the relative order in which the decoded pictures are displayed. Some of such coding schemes may cause the POC values to be reset (e.g., set to zero or set to some value signaled in the bitstream) whenever certain types of pictures appear in the bitstream. For example, the POC values of certain IRAP pictures may be reset, causing the POC values of other pictures preceding those IRAP pictures in decoding order to also be reset. This may be problematic when the IRAP pictures are not required to be aligned across different layers. For example, when one picture ("picA") is an IRAP picture and another picture ("picB") in the same access unit is not an IRAP picture, the POC value of a picture ("picC"), which is reset due to picA being an IRAP picture, in the layer containing picA may be different from the POC value of a picture ("picD"), which is not reset, in the layer containing picB, where picC and picD are in the same access unit. This causes picC and picD to have different POC values even though they belong to the same access unit (e.g., same output time). Thus, in this

example, the derivation process for deriving the POC values of picC and picD can be modified to produce POC values that are consistent with the definition of POC values and access units.

Picture Order Count (POC)

[00135] As discussed above, the value of a picture order count (POC) (e.g., PicOrderCntVal in HEVC) for a particular coded picture denotes the relative order of the particular coded picture in the picture output process with respect to other pictures in the same coded video sequence. In some embodiments, the POC comprises least significant bits (LSB) and most significant bits (MSB), and the POC may be obtained by concatenating the MSB and the LSB. In other embodiments, the POC may be obtained by adding the MSB value and the LSB value. The LSB may be signaled in the slice header, and the MSB may be computed by the encoder or the decoder based on the NAL unit type of the current picture and the MSB and LSB of one or more previous pictures in decoding order that are (1) not RASL or RADL pictures, (2) not discardable (e.g., pictures marked as "discardable," indicating that no other picture depends on them, thereby allowing them to be dropped to satisfy bandwidth constraints), (3) not sub-layer non-reference pictures (e.g., pictures that are not used for reference by other pictures in the same temporal sub-layer or the same layer), (4) has a temporal ID (e.g., temporal sub-layer ID) equal to 0. Such pictures described in (1)-(4) may be referred to herein as POC-anchor pictures. Similarly, pictures having a temporal ID value greater than 0, RASL or RADL pictures, discardable pictures, or sub-layer non-reference pictures may be referred to as non-POC-anchor pictures. POC-anchor pictures may further include pictures that an encoder and/or a decoder may not elect to remove from the bitstream (e.g., to satisfy a bandwidth constraint). POC-anchor pictures may further include any picture other than the types of pictures that an encoder and/or a decoder may be configured to remove from the bitstream (e.g., to satisfy a bandwidth constraint). Non-POC-anchor pictures may include any picture that is not a POC-anchor picture.

[00136] When the current picture is (1) an IRAP picture with NoRaslOutputFlag (e.g., a flag that indicates that RASL pictures are not to be output if set to 1 and indicates that RASL pictures are to be output if set to 0) equal to 1, or (2) a CRA picture that is the first picture of the bitstream, the value of POC MSB is inferred to be equal to 0. As described above, in a multi-layer bitstream (e.g., SHVC or MV-HEVC bitstream with more than one layer), there may exist access units (AU) where one or more pictures are IRAP pictures and one or more other pictures are non-IRAP pictures, and such AUs may be referred to as "non-

aligned IRAP AUs." When decoding bitstreams containing non-aligned IRAP AUs, it is possible (and likely) that the POCs derived based on the POC LSB values signaled in the bitstream would violate the bitstream conformance requirement that all pictures in an access unit should have the same POC value.

Layer Initialization Picture (LIP)

[00137] In some coding schemes, a layer initialization picture ("LIP picture") may be defined as a picture that is an IRAP picture that has a NoRaslOutputFlag flag (e.g., a flag that indicates that RASL pictures are not to be output if set to 1 and indicates that RASL pictures are to be output if set to 0) set to 1 or a picture that is contained an initial IRAP access unit, which is an IRAP access unit in which the base layer picture (e.g., a picture having a layer ID of 0 or smallest layer ID defined in the bitstream) has the NoRaslOutputFlag set to 1.

[00138] In some embodiments, an SPS can be activated at each LIP picture. For example, each IRAP picture that has a NoRaslOutputFlag flag set to 1 or each picture that is contained in an initial IRAP access unit, a new SPS, which may be different (e.g., specifying different picture resolutions, etc.) from the SPS that was previously activated. However, in a case where the LIP picture is not an IRAP picture (e.g., any picture contained in an initial IRAP access unit) and the base layer picture in the initial IRAP access unit is an IDR picture with a flag NoClrasOutputFlag flag (e.g., a flag that indicates that cross-layer random access skip pictures are not to be output if set to 1 and indicates that cross-layer random access skip pictures are to be output if set to 0) set to 0, the LIP picture should not be allowed to activate a new SPS. If a new SPS is activated at such the LIP picture in such a case, particularly when the contents of the SPS RBSP of the new SPS is different from that of the SPS that was previously active prior to the initial IRAP access unit, there could be problems in differing picture resolutions and error resilience. For example, the new SPS may update the resolution and use temporal prediction to refer to pictures of different sizes.

Bumping and Flushing of Pictures

[00139] Pictures that are decoded (e.g., so that they can be displayed or used to predict other pictures) are stored in a decoded picture buffer (DPB). The pictures that are to be output may be marked as "needed for output," and the pictures that are to be used to predict other pictures may be marked as "used for reference." Decoded pictures that are neither marked as "needed for output" nor as "used for reference" (e.g., pictures that were initially marked as "used for reference" or "needed for output" but subsequently marked as "not used

for reference" or "not needed for output") may be present in the DPB until they are removed by the decoding process. In output order conformant decoders, the process of removing pictures from the DPB often immediately follows the output of pictures that are marked as "needed for output." This process of output and subsequent removal may be referred to as "bumping."

[00140] There are also situations where the decoder may remove the pictures in the DPB without output, even though these pictures may be marked as "needed for output." For ease of description herein, decoded pictures that are present in the DPB at the time of decoding an IRAP picture (regardless of whether the decoded pictures are marked as "needed for output" or "used for reference") are referred to as "lagging DPB pictures" associated with the IRAP picture or "associated lagging DPB pictures" of the IRAP picture. Some examples of such situations, in the HEVC context, are described below.

[00141] In one example, when a CRA picture with NoRaslOutputFlag equal to a value of "1" is present in the middle of a bitstream (e.g., not the first picture in the bitstream), the lagging DPB pictures associated with the CRA picture would not be output and would be removed from the DPB. Such situations are likely to occur at splice points, where two bitstreams are joined together and the first picture of the latter bitstream is a CRA picture with NoRaslOutputFlag equal to a value of "1". In another example, when an IRAP picture picA that has NoRaslOutputFlag equal to a value of "1" and that is not a CRA picture (e.g., an IDR picture) is present in the middle of a bitstream and the resolution of the picture changes at picA (e.g., with the activation of a new SPS), the associated lagging DPB pictures of picA may be removed from the DPB before they can be output, because if the associated lagging DPB pictures continue to occupy the DPB, decoding of the pictures starting from picA may become problematic, for example, due to buffer overflow. In this case, the value of no_output_of_prior_pics_flag (e.g., a flag that indicates that pictures that were previously decoded and stored in the DPB should be removed from the DPB without being output if set to 1, and indicates that pictures that were previously decoded and stored in the DPB should not be removed from the DPB without being output if set to 0) associated with picA should be set equal to a value of "1" by the encoder or splicer, or NoOutputOfPriorPicsFlag (e.g., a derived value that may be determined based on the information included in the bitstream) may be derived to be equal to a value of "1" by the decoder, to flush the lagging pictures without output out of the DPB. The splicing operation is described further below with respect to **FIG. 4**.

[00142] This process of removing associated lagging DPB pictures from the DPB without output may be referred to as "flushing." Even in situations not described above, an IRAP picture may specify the value of no_output_of_prior_pics_flag equal to a value of "1", so that the decoder will flush the associated DPB lagging pictures of the IRAP picture.

Bitstream including a Splice Point

[00143] With reference to **FIG. 4**, an example bitstream having a splice point will be described. **FIG. 4** shows a multi-layer bitstream 400 created by splicing bitstreams 410 and 420. The bitstream 410 includes an enhancement layer (EL) 410A and a base layer (BL) 410B, and the bitstream 420 includes an EL 420A and a BL 420B. The EL 410A includes an EL picture 412A, and the BL 410B includes a BL picture 412B. The EL 420A includes EL pictures 422A, 424A, and 426A, and the BL 420B includes BL pictures 422B, 424B, and 426B. The multi-layer bitstream 400 further includes access units (AUs) 430-460. The AU 430 includes the EL picture 412A and the BL picture 412B, the AU 440 includes the EL picture 422A and the BL picture 422B, the AU 450 includes the EL picture 424A and the BL picture 424B, and the AU 460 includes the EL picture 426A and the BL picture 426B. In the example of **FIG. 4**, the BL picture 422B is an IRAP picture, and the corresponding EL picture 422A in the AU 440 is a trailing picture (e.g., a non-IRAP picture), and consequently, the AU 440 is a non-aligned IRAP AU. Also, it should be noted that the AU 440 is an access unit that immediately follows a splice point 470.

[00144] Although the example of **FIG. 4** illustrates a case where two different bitstreams are joined together, in some embodiments, a splice point may be present when a portion of the bitstream is removed. For example, a bitstream may have portions A, B, and C, portion B being between portions A and C. If portion B is removed from the bitstream, the remaining portions A and C may be joined together, and the point at which they are joined together may be referred to as a splice point. More generally, a splice point as discussed in the present application may be deemed to be present when one or more signaled or derived parameters or flags have predetermined values. For example, without receiving a specific indication that a splice point exists at a particular location, a decoder may determine the value of a flag (e.g., NoClrasOutputFlag), and perform one or more techniques described in this application based on the value of the flag.

Support for Reference Layer and Enhancement Layer Coded Using Different Standards

[00145] In some cases of multi-layer coding, the reference layer (RL) may be coded using one standard and the enhancement layer (EL) may be coded using another standard. For example, the RL may be coded according to H.264/AVC, and the EL may be coded according to H.265/HEVC. Providing support for a RL that is coded using an earlier version of the standard can be beneficial since video data coded using the earlier version may be utilized in multi-layer video coding using the current standard. However, managing many aspects of processing the RL coded in a different standard can cause support for multi-layer coding to become complicated. For example, the encoder or decoder may need to handle output of RL pictures, maintain decoded picture stores for RL pictures, etc. To facilitate discussion, coding RL and EL using different standards may be referred to as multi-layer video coding using different standards. For illustrative purposes, the description refers to using different standards, but the techniques described herein can also apply to using different versions of the same standard. The techniques may also apply to using different coding schemes or using different versions of a coding scheme.

[00146] In order to address these and other challenges, the techniques according to certain aspects can support multi-layer video coding using different standards in a simplified manner. For instance, the HEVC decoder may minimize managing and processing of RL pictures coded using a standard other than HEVC. For illustrative purposes, the RL will be explained as being coded using H.264/AVC, and the EL will be will be explained as being coded using H.265/HEVC. However, any combination of different standards can be used to code the RL and EL. According to certain aspects, the techniques minimize managing and processing of RL pictures as follows: (1) the decoded RL pictures are provided by external means (e.g., no encapsulation is provided), and (2) the output of RL pictures, including the synchronization with output of EL pictures, are controlled by external means. Encapsulation may refer to defining certain NAL unit header syntax for RL pictures such that RL pictures are provided in the bitstream using the specific NAL unit header syntax. External means may refer to a coder (e.g., an encoder or a decoder) that supports the standard used to code the RL. The HEVC decoder can implement certain rules, explained in detail below, in order to support multi-layer video coding using different standards. As explained above, the techniques can also apply to using different versions of a standard, not simply using different

standards, and also apply to using different coding schemes or using different versions of a coding scheme.

[00147] Various terms used throughout this disclosure are broad terms having their ordinary meaning. In addition, in some embodiments, certain terms relate to the following video concepts. A standard can refer to a video coding standard, such as H.264/AVC, H.265/HEVC, etc. In some embodiments, a standard can refer to different extensions of the same standard, such as H.265/HEVC, SHVC, MV-HEVC, etc. For example, the reference layer is encoded using HEVC, and the enhancement layer is encoded using SHVC or MV-HEVC. As explained above, external means in this context may refer to any apparatus or entity that is not a part of the EL decoder but interacts with the EL decoder, e.g., through an application programming interface (API). In one example, external means may refer to a coder (e.g., an encoder or a decoder) that supports the standard used to code the RL. In another example, external means may refer to a part of the receiver that contains the EL decoder. For instance, the same receiver may contain both the EL decoder and the external means. In such case, the external means is still external to the EL decoder in the sense that it is not a part of the EL decoder. In certain embodiments, external means may also be referred to as an external apparatus.

Example Embodiment

[00148] In one embodiment, the following rules are implemented in order to support multi-layer coding using different standards:

1) For the current access unit, either no information is provided (meaning no base layer picture is present for the current access unit) or the following information of the base layer picture is provided by external means:
   - The decoded sample values of the base layer decoded picture
   - The representation format of the base layer decoded picture, including the width and height in luma samples, the colour format, the luma bit depth, and the chroma bit depth.
   - Whether the base layer picture is an IDR picture or not.
   - Optionally, whether the picture is a frame or a field, and when a field, the field parity (indicating whether the field is a top field or a bottom field). If not provided, the decoded picture is inferred to be a frame picture.
2) Output of base layer pictures is the responsibility of the base layer decoder. Optionally, only one non-base layer is present and it is the only target output layer.

3) The (multi-standard) SHVC decoder would only need to keep one decoded picture store of memory for a base layer decoded picture, and this memory is not considered as part of the DPB.
4) Association of a base layer decoded picture to an access unit is the responsibility of external means (e.g. the base layer decoder or other external means).
5) The nuh_layer_id of the base layer decoded picture is set equal to 0. Alternatively, the nuh_layer_id of the base layer decoded picture is set equal to the nuh_layer_id of the entry in the inter-layer reference picture set of an enhancement layer picture with the lowest value of nuh_layer_id among all the enhancement layer pictures in the access unit.
6) The picture order count (POC) of the base layer decoded picture is set equal to the picture order count of the enhancement layer pictures. Note that in this case the actual picture order count of a base layer picture decoded by the base layer decoder in such a scalable or multiview codec might be different than the picture order count value of the same picture when it is decoded by an AVC decoder.
7) The base layer decoded picture is marked as "used for long-term reference."
8) For the coded picture buffer operations of the hypothetical reference decoder or buffering model, the base layer is considered as having zero bits.
9) For decoded picture buffer operations of the hypothetical reference decoder or buffering model, only decoded pictures of enhancement layers are considered.

[00149] The rules can be implemented by the coder (e.g., encoder and/or decoder) that codes the EL. Some or all of the rules may also be implemented by the coder that codes the RL. For example, the RL coder can be modified to provide the information about the RL pictures to the EL coder. Any functionality indicated as being performed by the RL coder can be performed by any other external means, and any functionality indicated as being performed by an external means can be performed by the RL coder. The rules are explained in terms of a base layer and an enhancement layer above, but the rules can apply to any enhancement layer and its corresponding reference layer, which may or may not be a base layer. Each rule is explained in turn below.

Rule 1 – Information of RL Pictures

[00150] If the current AU does not include a RL picture, the external means does not provide any information about RL pictures. If the current AU includes a RL picture, the external means provides the following three types of information:

(1) decoded sample values of the RL decoded picture;

(2) representation format of the RL decoded picture, including the width and height in luma samples, the color format, the luma bit depth, and the chroma bit depth;

(3) whether the RL picture is an IDR picture or not.

The external means can provide only (1) through (3) about the RL picture in order to support multi-layer coding using different standards. The information (3) can be important because decoding of EL pictures and output of EL pictures may depend on (3) in some cases.

[00151] The external means may optionally provide the following type of information:

(4) whether the picture is a frame or a field, and when the picture is a field, the field parity (indicating whether the field is a top field or a bottom field).

[00152] A frame can refer to a picture that is a progressive frame. A field can refer to a picture that is an interlace picture, which comprises half of the sample rows, either the odd or even rows, of a frame. Field parity can indicate whether a field is a top field or a bottom field. A top field can refer to the field that comprises the even number of samples rows, with the row indexing number starting from 0. A bottom field can refer to the field that comprises the odd number of samples rows, with the row indexing number starting from 0. If (4) is not provided, the decoded RL picture is inferred to be a frame picture.

[00153] Any of the information (1) through (4) can be provided using Application Programing Interfaces (APIs) or other appropriate methods. In one example, the RL coder can have an API to pass the decoded RL pictures, and the EL coder can have an API to accept the decoded RL pictures. The two APIs can be connected in order to send the decoded RL picture values from the RL coder to the EL coder. In another example, the RL coder can store the decoded RL pictures in a file or a buffer, and the EL coder can access the decoded RL pictures from the file or the buffer.

Rule 2 – Output of RL Pictures

[00154] The RL coder or another external means handles the output of RL pictures. Optionally, it is specified that only one non-reference layer (e.g., one EL) is present, and it is the only target output layer. Accordingly, the EL coder does not need to output any RL pictures.

Rule 3 – Decoded Picture Store for RL Decoded Pictures

[00155] The EL coder can keep just one decoded picture store of memory for a RL decoded picture, and the decoded picture store for RL decoded pictures are not considered a

part of the DPB. Decoded RL pictures are generally only used in inter-layer prediction (ILP) within the same AU, so once the EL coder completes decoding the current AU, decoded RL pictures for the current AU can be removed from the decoded picture store. Therefore, the memory size of the decoded picture store can remain fixed, and there can be one frame buffer size for the decoded picture store for RL pictures.

Rule 4 – Association of RL Decoded Picture to Access Unit

[00156] The RL coder or another external means handles the association of a RL decoded picture to an AU. For example, each EL picture belongs to a particular AU, and the RL coder or another external means determines how to match the RL decoded picture to an AU in the EL. In one embodiment, the RL decoded picture can be matched to an AU based on timestamps. The bitstream for the EL and the bitstream for the BL both include timestamps, and the RL coder or another external means matches the RL decoded picture to an AU in the EL based on the timestamps.

Rule 5 – Setting of Layer ID of the RL Decoded Picture

[00157] The EL coder sets the nuh_layer_id of the RL decoded picture to 0. The variable nuh_layer_id can refer to the identifier of a layer in multi-layer coding. Alternatively, the EL coder sets the nuh_layer_id of the RL decoded picture to the nuh_layer_id of the entry in the inter-layer reference picture set of an EL picture that has the lowest value of nuh_layer_id among all the EL pictures in the AU. For example, if there are one or more lower layers other than the EL that can be used for ILP, the EL coder sets the nuh_layer_id of the RL decoded picture to the lowest nuh_layer_id value from these layers.

Rule 6 – Setting of Picture Order Count of the RL Decoded Picture

[00158] The EL coder sets the POC of the RL decoded picture equal to the POC of the enhancement layer pictures. The POC of the RL decoded picture may be different when decoded by the EL coder and when decoded by the RL coder. The EL coder may or may not have access to the POC of the RL decoded picture since the EL coder only receives the decoded sample values of the RL decoded picture. The EL coder may not necessarily have to know the original POC value of the picture obtained by the RL coder. The EL coder can simply set the POC of the RL decoded picture equal to the POC of the EL picture.

Rule 7 – Marking of RL Decoded Picture

[00159] The EL coder marks the RL decoded picture as "used for long-term reference." If a RL decoded picture is not marked as "used for long-term reference," certain scaling operations may be performed for ILP, which can involve a division operation by the delta of POC values of the RL picture and the EL picture. Since the EL coder sets the POC of the RL decoded picture to be the same as the POC of the EL picture, this would involve dividing by 0. Accordingly, in order to prevent a division operation by 0, the EL coder can mark the RL decoded picture as "used for long-term reference."

Rule 8 – Coded Picture Buffer Operations

[00160] For the coded picture buffer (CPB) operations of the hypothetical reference decoder (HRD) or buffering model, the EL coder considers the RL as having zero bits. HRD can refer to a buffering model for receiving a bitstream from a network connection and decoding the bitstream. For instance, the bitstream includes coded pictures, which are compressed pictures, and the coded pictures are stored in the CPB for decoding. Since the EL coder receives the decoded sample values of the RL decoded picture from external means, the EL coder does not need to take into account the coded pictures for the RL picture with respect to CPB operations.

Rule 9 – Decoded Picture Buffer Operations

[00161] For decoded picture buffer operations of the HRD or buffering model, the EL coder only considers decoded pictures of enhancement layers. As mentioned above, the RL decoded pictures are not decoded by the EL coder, and are stored in a separate decoded picture store that is not considered to be a part of the DPB. Therefore, the EL coder does not take into account the decoded RL pictures with respect to DPB operations.

Method of Supporting Multi-Layer Coding Using Different Standards

[00162] FIG. 5 is a flowchart illustrating a method of coding video information, according to one embodiment of the present disclosure. The method relates to providing support for multi-layer coding using different standards. The process 500 may be performed by an encoder (e.g., the encoder as shown in FIG. 2A, 2B, etc.), a decoder (e.g., the decoder as shown in FIG. 3A, 3B, etc.), or any other component, depending on the embodiment. The blocks of the process 500 are described with respect to the decoder 33 in FIG. 3B, but the

process 500 may be performed by other components, such as an encoder, as mentioned above. The layer 1 video decoder 30B of the decoder 33 and/or the layer 0 decoder 30A of the decoder 33 may perform the process 500, depending on the embodiment. All embodiments described with respect to FIG. 5 may be implemented separately, or in combination with one another. Certain details relating to the process 500 are explained above and below, e.g., with respect to FIG. 4.

[00163] The process 500 starts at block 501. The decoder 33 can include a memory for storing video information associated with an enhancement layer (EL) and a corresponding reference layer (RL).

[00164] At block 502, the decoder 33 obtains information associated with a reference layer picture in the current access unit (AU). The RL is coded using a standard (e.g., H.264/AVC). The standard used to code the RL may also be referred to as the "RL standard." The information associated with the RL picture can consist of: (1) decoded sample values of the RL picture; (2) the representation format of the RL picture; and (3) an indication of whether the RL picture is an instantaneous decoding refresh (IDR) picture or not. The information associated with the RL picture can be provided by an external means or apparatus, for example, external to the decoder 33. In one embodiment, the external means is a coder that is configured to code video information using the RL standard. For example, the coder can be the coder that was used to code the RL or any other external means. The representation format of the RL picture can include at least one of: the width of the RL picture, the height of the RL picture, the color format of the RL picture, the bit depth of luma components of the RL picture, or the bit depth of chroma components of the RL picture. The width of the RL picture and the height of the RL picture may be specified in units of luma samples. In one embodiment, the information is provided using one or more application programming interfaces (APIs). In other embodiments, the information is provided using a file, a buffer, or other appropriate means.

[00165] In some embodiments, the decoder 33 receives additional information associated with the RL picture in the current AU. The additional information can include (4) an indication of whether the RL picture is a frame or a field. If the RL picture is a field, the additional information can further include: (5) the field parity. The additional information can be optional.

[00166] At block 503, the decoder 33 codes the EL picture in the current AU based on the obtained information. The EL picture is coded using a standard that is different from the standard used to code the RL (e.g., H.265/HEVC or SHVC). In one embodiment, the

standard used to code the RL is H.264/AVC and the standard used to code the EL is H.265/HEVC or SHVC. The standard used to code the EL may be referred to as the "EL standard."

[00167] In certain embodiments, the decoder 33 may implement one or more of the following features:

- The decoder 33 does not handle the output of RL decoded pictures. For example, the decoder 33 is configured to not output the decoded sample values of the RL picture.
- The memory of the decoder 33 includes a decoded picture buffer (DPB), and the decoder 33 stores the decoded sample values of the RL picture in a decoded picture store that is not a part of the DPB.
- The decoder 33 does not handle association of RL pictures to an AU in the EL. For example, the decoder 33 obtains the AU of the RL picture from the coder that is configured to code video information using the RL standard. The coder may associate the RL picture to an AU in the EL based on timestamps in the EL bitstream and the BL bitstream, and provide the AU of the RL picture when requested by the decoder 33.
- The decoder 33 sets the layer identifier (e.g., nuh_layer_ID) of the RL picture equal to the layer identifier of the EL picture.
- The decoder 33 sets the picture order count (POC) of the RL picture equal to the POC of the EL picture.
- The decoder 33 marks the RL picture as a long-term reference picture (e.g., "used for long-term reference").
- The memory of the decoder 33 includes a coded picture buffer (CPB), and the decoder 33 considers the RL to have 0 bits for CPB operations. For example, the decoder 33 sets the RL to have 0 bits in CPB operations.
- The decoder 33 considers only decoded pictures of the EL for DPB operations. For example, the decoder 33 is configured to not refer to the decoded sample values of the RL picture in DPB operations.

[00168] The process 500 ends at block 504. Blocks may be added and/or omitted in the process 500, depending on the embodiment, and blocks of the process 500 may be performed in different orders, depending on the embodiment.

[00169] Any features and/or embodiments described with respect to support for multi-layer coding using different standards in this disclosure may be implemented separately or in

any combination thereof. For example, any features and/or embodiments described in connection with FIGS. 1-4 and other parts of the disclosure may be implemented in any combination with any features and/or embodiments described in connection with FIG. 5, and vice versa.

[00170] 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.

[00171] The various illustrative logical blocks, 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.

[00172] 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.

[00173] 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.

[00174] 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.

[00175] Various embodiments of the disclosure have been described. These and other embodiments are within the scope of the following claims.

WHAT IS CLAIMED IS:

1. An apparatus for coding video information, the apparatus comprising:

a memory configured to store video information associated with an enhancement layer (EL) and a corresponding reference layer (RL); and

a processor operationally coupled to the memory and configured to:

code an EL picture in a current access unit (AU), the EL coded using a first standard that is different from a second standard that is used to code the RL, wherein the coding of the EL picture is based on information associated with a RL picture in the current access unit, the information associated with the RL picture provided by an external means and consisting of:

(1) decoded sample values of the RL picture;

(2) a representation format of the RL picture; and

(3) an indication of whether the RL picture is an instantaneous decoding refresh (IDR) picture.

2. The apparatus of Claim 1, wherein the external means is a coder that is configured to code video information using the second standard.

3. The apparatus of Claim 1, wherein the processor is further configured to:

receive second information associated with the RL picture in the current AU, the second information comprising: (4) an indication of whether the RL picture is a frame or a field; and

code the EL picture based on the information associated with the RL picture in the current AU and the second information.

4. The apparatus of Claim 3, wherein if the RL picture is a field, the second information associated with the RL picture further comprises: (5) a field parity.

5. The apparatus of Claim 1, wherein the first standard is H.265/HEVC video coding standard and the second standard is H.264/AVC video coding standard.

6. The apparatus of Claim 1, wherein the processor is further configured to not output the decoded sample values of the RL picture.

7. The apparatus of Claim 6, wherein the memory includes a decoded picture buffer (DPB) and the processor is further configured to store the decoded sample values of the RL picture in a decoded picture store that is not a part of the decoded picture buffer.

8. The apparatus of Claim 7, wherein the processor is further configured to obtain from the external apparatus information relating to an access unit to which the RL picture belongs.

9. The apparatus of Claim 8, wherein the processor is further configured to set a layer identifier of the RL picture equal to a layer identifier of the EL picture.

10. The apparatus of Claim 9, wherein the processor is further configured to set a picture order count (POC) of the RL picture equal to a POC of the EL picture.

11. The apparatus of Claim 10, wherein the processor is further configured to mark the RL picture as a long-term reference picture.

12. The apparatus of Claim 11, wherein the memory includes a coded picture buffer (CPB) and the processor is further configured to set the RL to have 0 bits in CPB operations.

13. The apparatus of Claim 12, wherein the processor is further configured to not refer to the decoded sample values of the RL picture in DPB operations.

14. The apparatus of Claim 1, wherein the representation format of the RL picture comprises at least one of: a width of the RL picture, a height of the RL picture, a color format of the RL picture, a bit depth of luma components of the RL picture, or a bit depth of chroma components of the RL picture.

15. The apparatus of Claim 14, wherein the width of the RL picture and the height of the RL picture are specified in units of luma samples.

16. The apparatus of Claim 1, wherein the information associated with the RL picture is provided using one or more application programming interfaces (APIs).

17. The apparatus of Claim 1, wherein the apparatus is selected from a group consisting of one or more of: a desktop computer, a notebook computer, a laptop computer, a tablet computer, a set-top box, a telephone handset, a smart phone, a smart pad, a television, a camera, a display device, a digital media player, a video gaming console, and a video streaming device.

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

using one or more computing devices comprising computer hardware, storing video information associated with an enhancement layer (EL) and a corresponding reference layer (RL) in a memory; and

using the one or more computing devices comprising computer hardware, coding an EL picture in a current access unit (AU), the EL coded using a first standard that is different from a second standard that is used to code the RL, wherein

the coding of the EL picture is based on information associated with a RL picture in the current access unit, the information associated with the RL picture provided by an external means and consisting of:

(1) decoded sample values of the RL picture;

(2) a representation format of the RL picture; and

(3) an indication of whether the RL picture is an instantaneous decoding refresh (IDR) picture.

19. The method of Claim 18, wherein the one or more computing devices are configured to not output the decoded sample values of the RL picture.

20. The method of Claim 19, wherein the memory includes a decoded picture buffer (DPB) and the method further comprises storing the decoded sample values of the RL picture in a decoded picture store that is not a part of the decoded picture buffer.

21. The method of Claim 20, further comprising obtaining from the external means information relating to an access unit to which the RL picture belongs.

22. The method of Claim 21, further comprising setting a layer identifier of the RL picture equal to a layer identifier of the EL picture.

23. The method of Claim 22, further comprising setting a picture order count (POC) of the RL picture equal to a POC of the EL picture.

24. The method of Claim 23, further comprising marking the RL picture as a long-term reference picture.

25. The method of Claim 24, wherein the memory includes a coded picture buffer (CPB) and the method further comprises setting the RL to have 0 bits in CPB operations.

26. The method of Claim 25, wherein the one or more computing devices are configured to not refer to the decoded sample values of the RL picture in DPB operations.

27. The method of Claim 18, wherein the representation format of the RL picture comprises at least one of: a width of the RL picture, a height of the RL picture, a color format of the RL picture, a bit depth of luma components of the RL picture, or a bit depth of chroma components of the RL picture.

28. A non-transitory computer readable medium comprising instructions that when executed on a processor comprising computer hardware cause the processor to:

store video information associated with an enhancement layer (EL) and a corresponding reference layer (RL); and

code an EL picture in a current access unit (AU), the EL coded using a first standard that is different from a second standard that is used to code the RL, wherein

the coding of the EL picture is based on information associated with a RL picture in the current access unit, the information associated with the RL picture provided by an external means and consisting of:

(1) decoded sample values of the RL picture;

(2) a representation format of the RL picture; and

(3) an indication of whether the RL picture is an instantaneous decoding refresh (IDR) picture.

29. The computer readable medium of Claim 28, wherein the external means is a coder that is configured to code video information using the second standard.

30. An apparatus for coding video information, the apparatus comprising:

means for storing video information associated with an enhancement layer (EL) and a corresponding reference layer (RL); and

means for coding an EL picture in a current access unit (AU), the EL coded using a first standard that is different from a second standard that is used to code the RL, wherein the coding of the EL picture is based on information associated with a RL picture in the current access unit, the information associated with the RL picture provided by an external means and consisting of:

(1) decoded sample values of the RL picture;

(2) a representation format of the RL picture; and

(3) an indication of whether the RL picture is an instantaneous decoding refresh (IDR) picture.

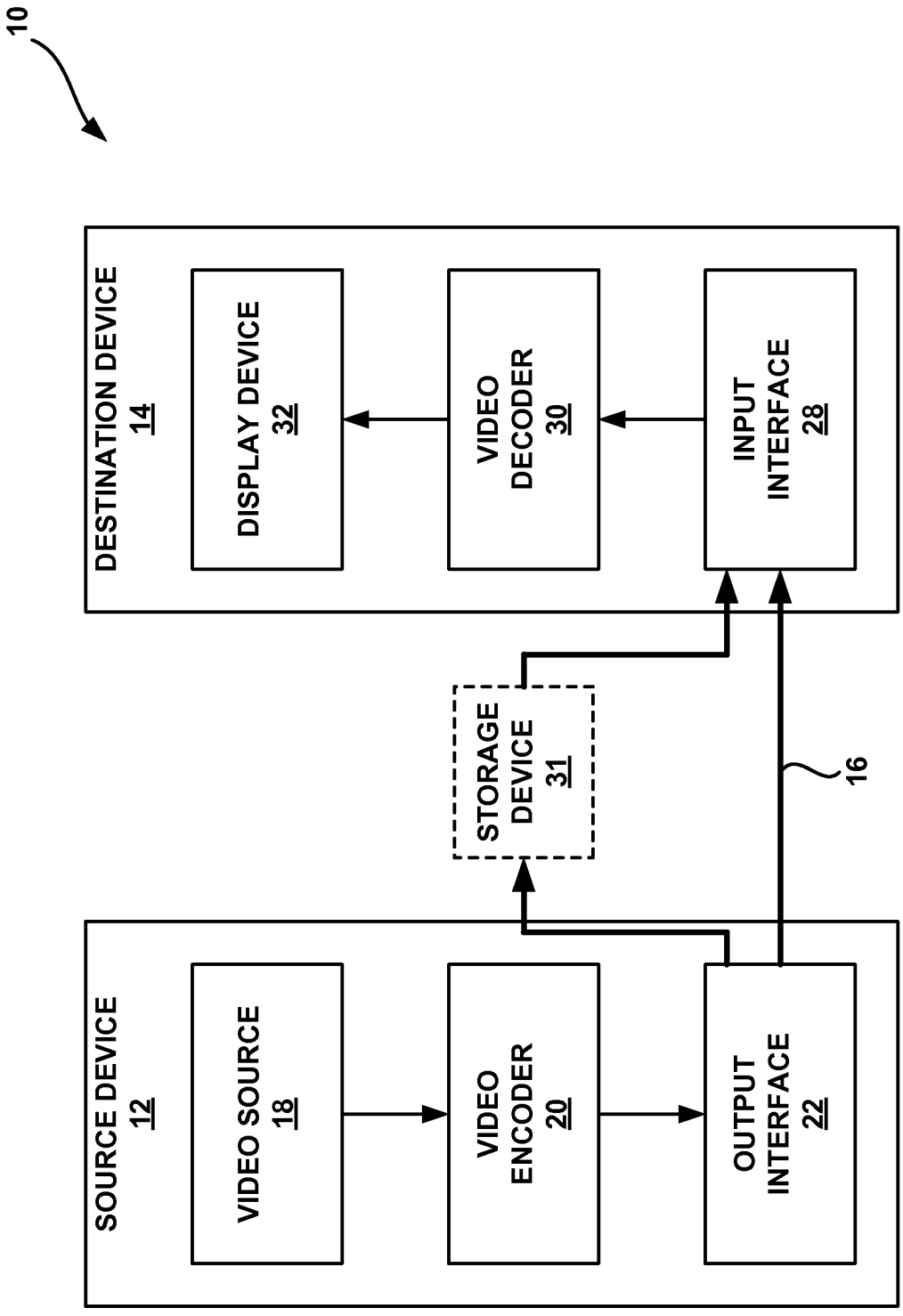
10
DESTINATION DEVICE
14
DISPLAY DEVICE
32
VIDEO
DECODER
30
INPUT
INTERFACE
28
STORAGE
DEVICE
31
16
SOURCE DEVICE
12
VIDEO SOURCE
18
VIDEO
ENCODER
20
OUTPUT
INTERFACE
22

FIG. 1A

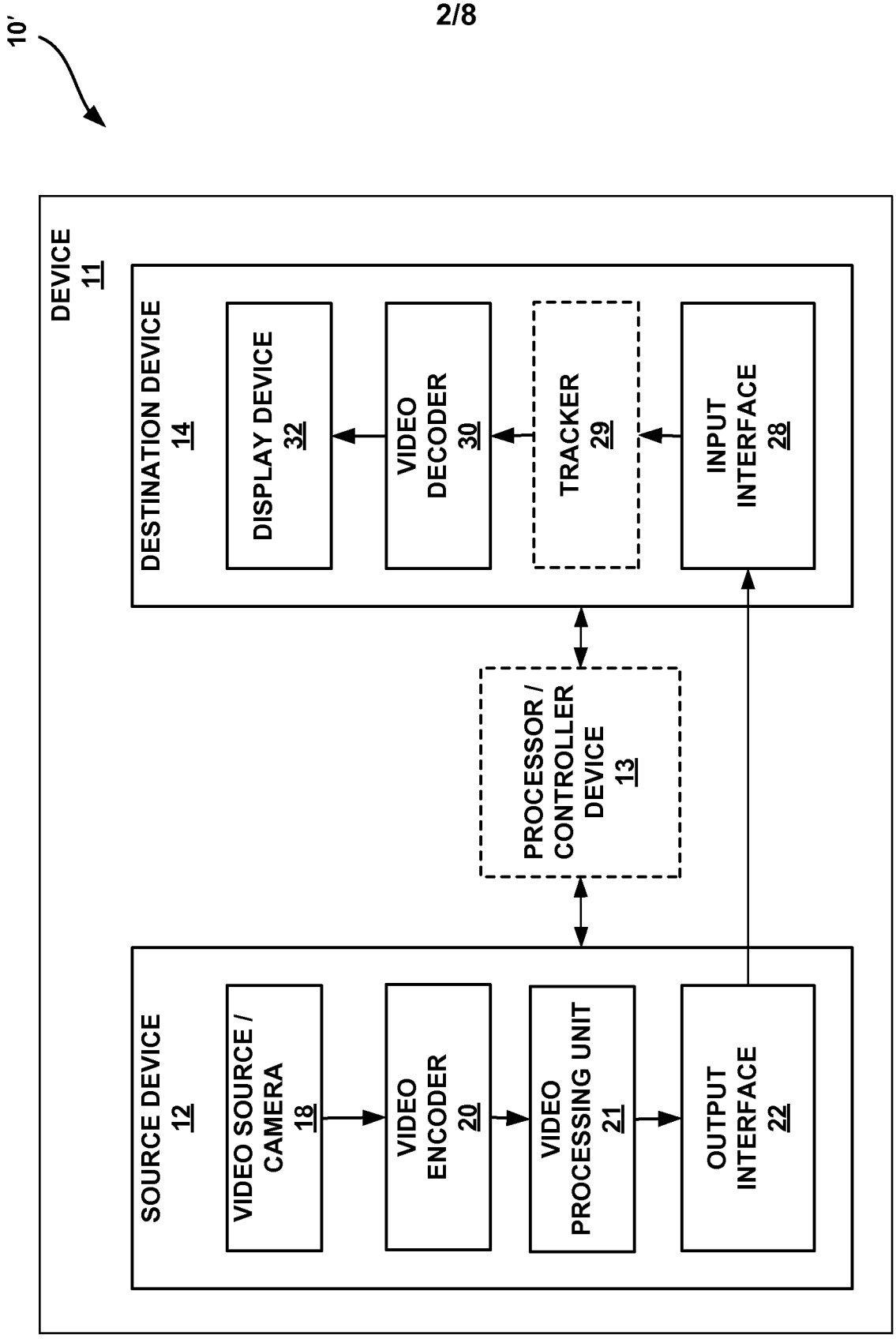
10'
DEVICE
11
DESTINATION DEVICE
14
DISPLAY DEVICE
32
VIDEO
DECODER
30
TRACKER
29
INPUT
INTERFACE
28
PROCESSOR /
CONTROLLER
DEVICE
13
SOURCE DEVICE
12
VIDEO SOURCE /
CAMERA
18
VIDEO
ENCODER
20
VIDEO
PROCESSING UNIT
21
OUTPUT
INTERFACE
22

FIG. 1B

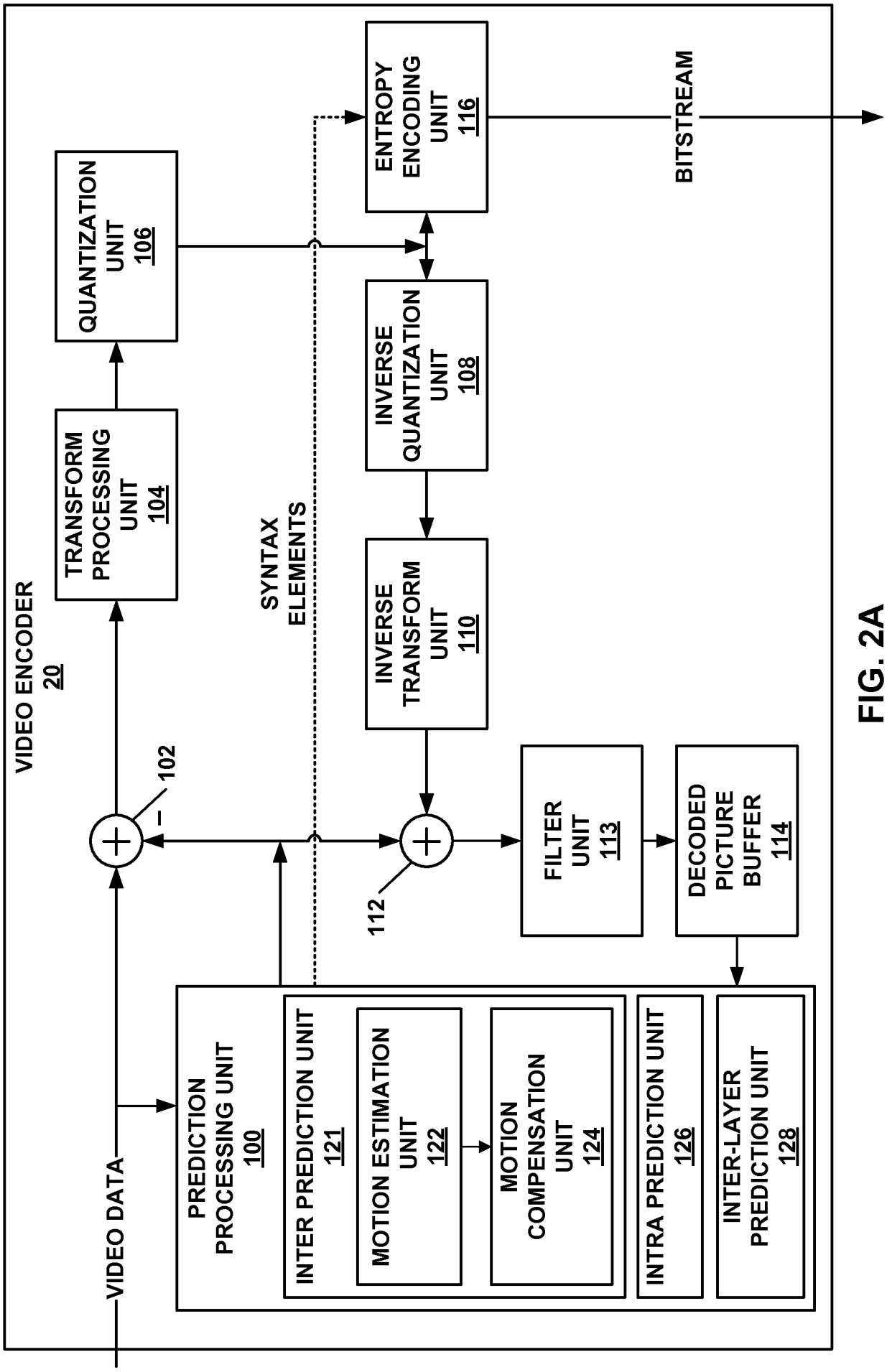
VIDEO ENCODER
20
VIDEO DATA
PREDICTION PROCESSING UNIT
100
INTER PREDICTION UNIT
121
MOTION ESTIMATION UNIT
122
MOTION COMPENSATION UNIT
124
INTRA PREDICTION UNIT
126
INTER-LAYER PREDICTION UNIT
128
102
TRANSFORM PROCESSING UNIT
104
QUANTIZATION UNIT
106
SYNTAX ELEMENTS
INVERSE QUANTIZATION UNIT
108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

FIG. 2A

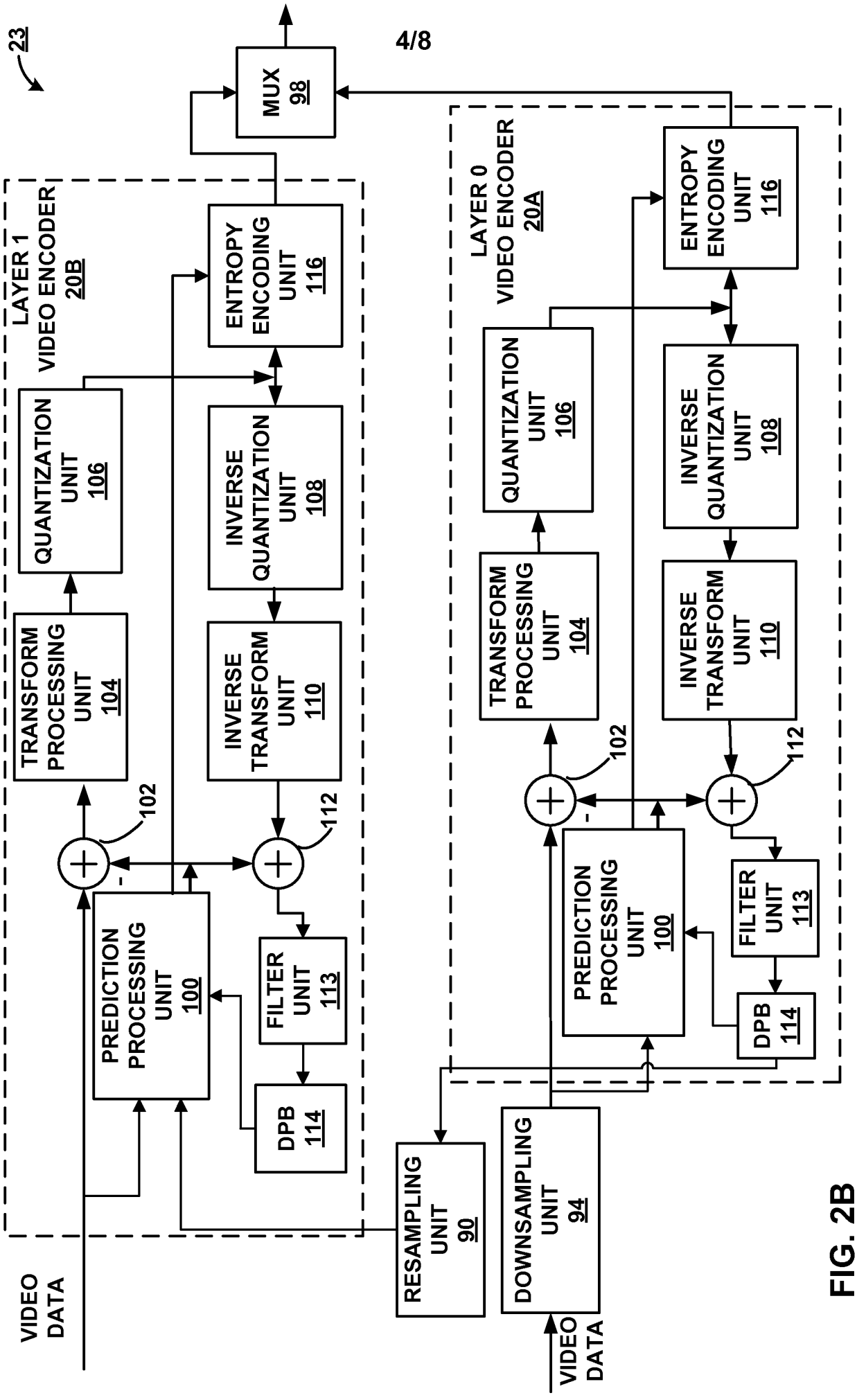
23
MUX 98
LAYER 1 VIDEO ENCODER 20B
ENTROPY ENCODING UNIT 116
QUANTIZATION UNIT 106
INVERSE QUANTIZATION UNIT 108
TRANSFORM PROCESSING UNIT 104
INVERSE TRANSFORM UNIT 110
102
112
PREDICTION PROCESSING UNIT 100
FILTER UNIT 113
DPB 114
VIDEO DATA
RESAMPLING UNIT 90
LAYER 0 VIDEO ENCODER 20A
ENTROPY ENCODING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

FIG. 2B

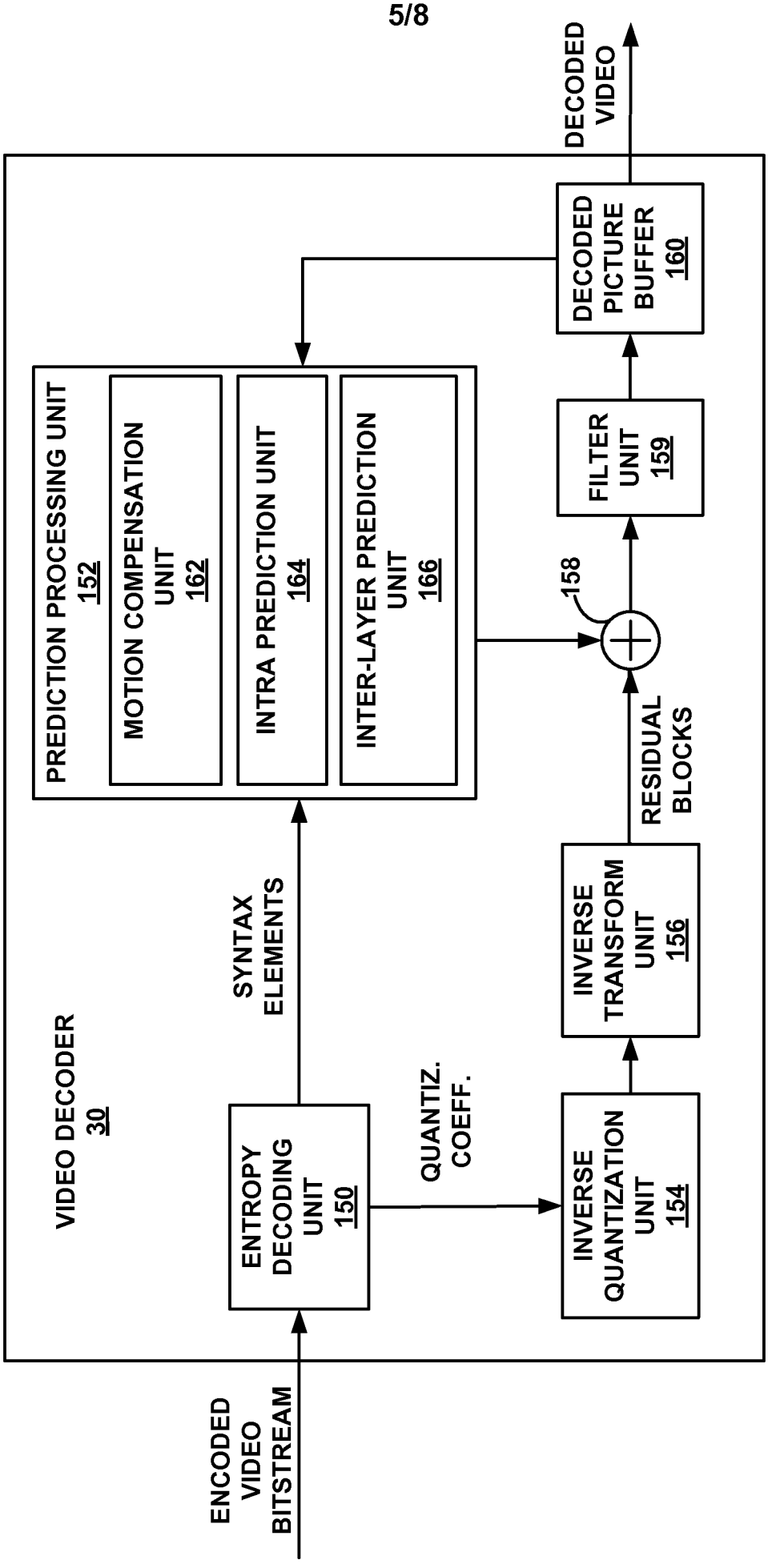
ENCODED VIDEO BITSTREAM
VIDEO DECODER 30
ENTROPY DECODING UNIT 150
SYNTAX ELEMENTS
QUANTIZ. COEFF.
INVERSE QUANTIZATION UNIT 154
INVERSE TRANSFORM UNIT 156
RESIDUAL BLOCKS
158
PREDICTION PROCESSING UNIT 152
MOTION COMPENSATION UNIT 162
INTRA PREDICTION UNIT 164
INTER-LAYER PREDICTION UNIT 166
FILTER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

FIG. 3A

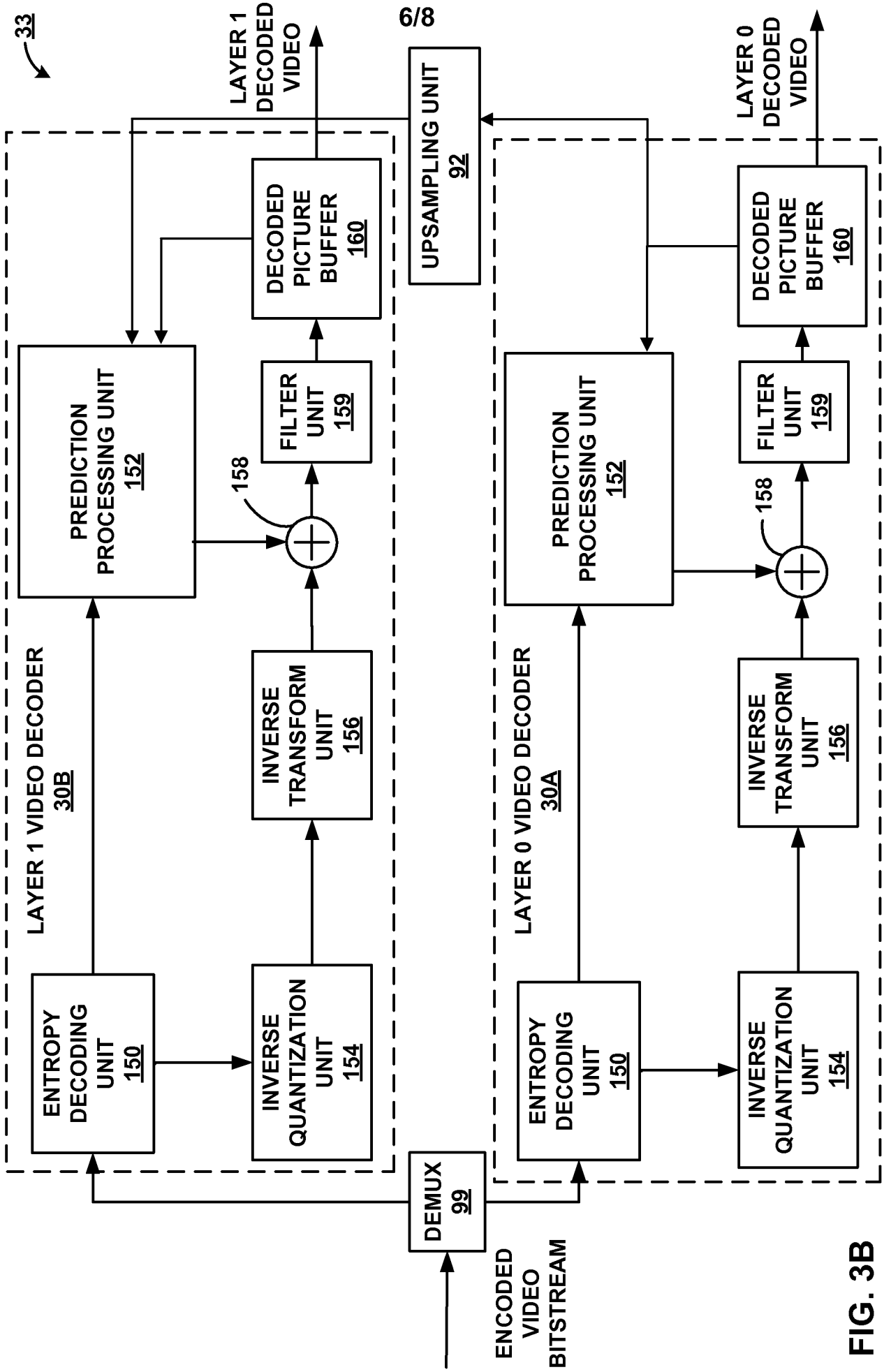
33
LAYER 1 DECODED VIDEO
UPSAMPLING UNIT 92
LAYER 0 DECODED VIDEO
DECODED PICTURE BUFFER 160
FILTER UNIT 159
PREDICTION PROCESSING UNIT 152
158
INVERSE TRANSFORM UNIT 156
INVERSE QUANTIZATION UNIT 154
ENTROPY DECODING UNIT 150
LAYER 1 VIDEO DECODER 30B
LAYER 0 VIDEO DECODER 30A
DEMUX 99
ENCODED VIDEO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

FIG. 3B

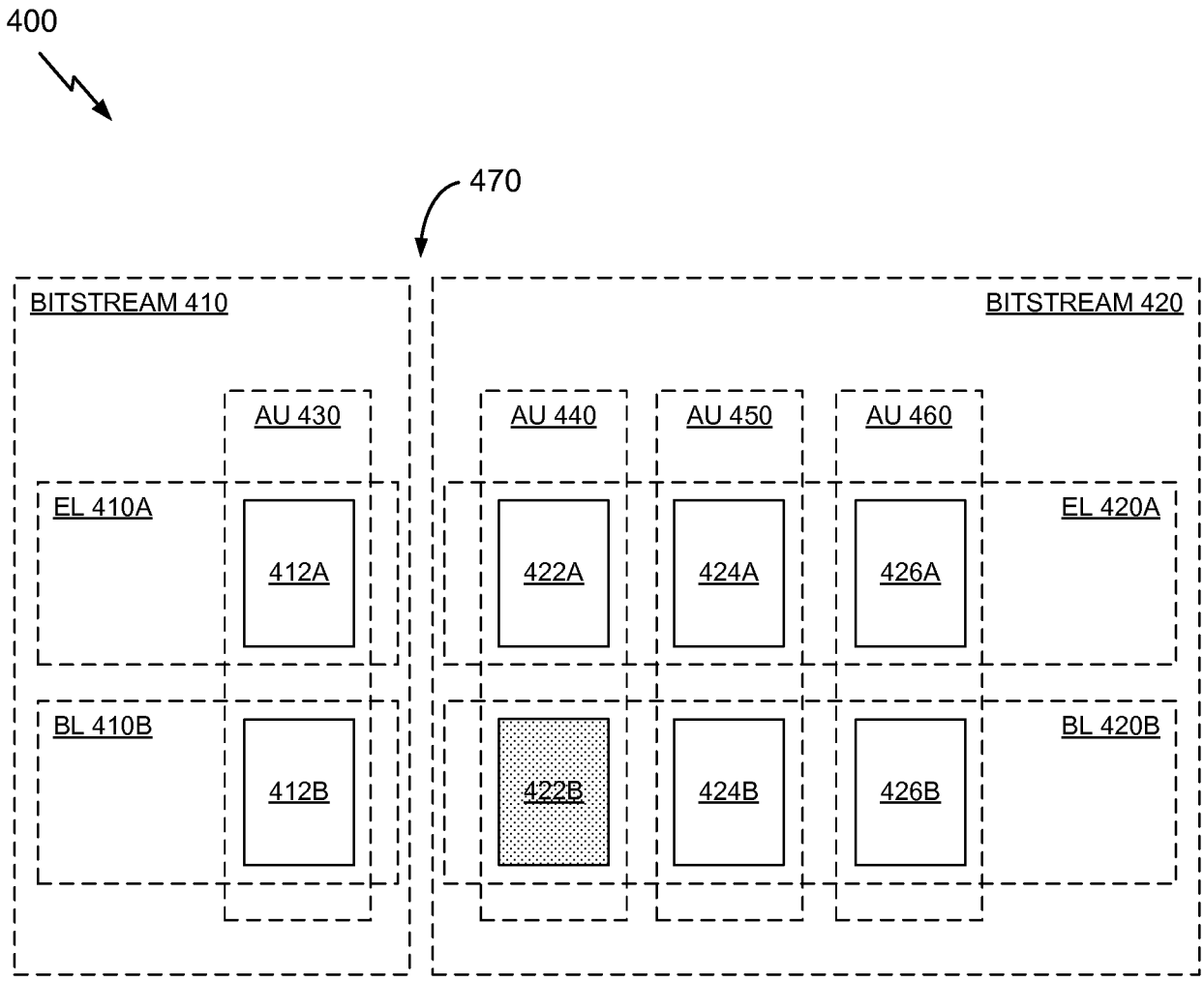
400
470
BITSTREAM 410
BITSTREAM 420
AU 430
AU 440
AU 450
AU 460
EL 410A
EL 420A
412A
422A
424A
426A
BL 410B
BL 420B
412B
422B
424B
426B

FIG. 4

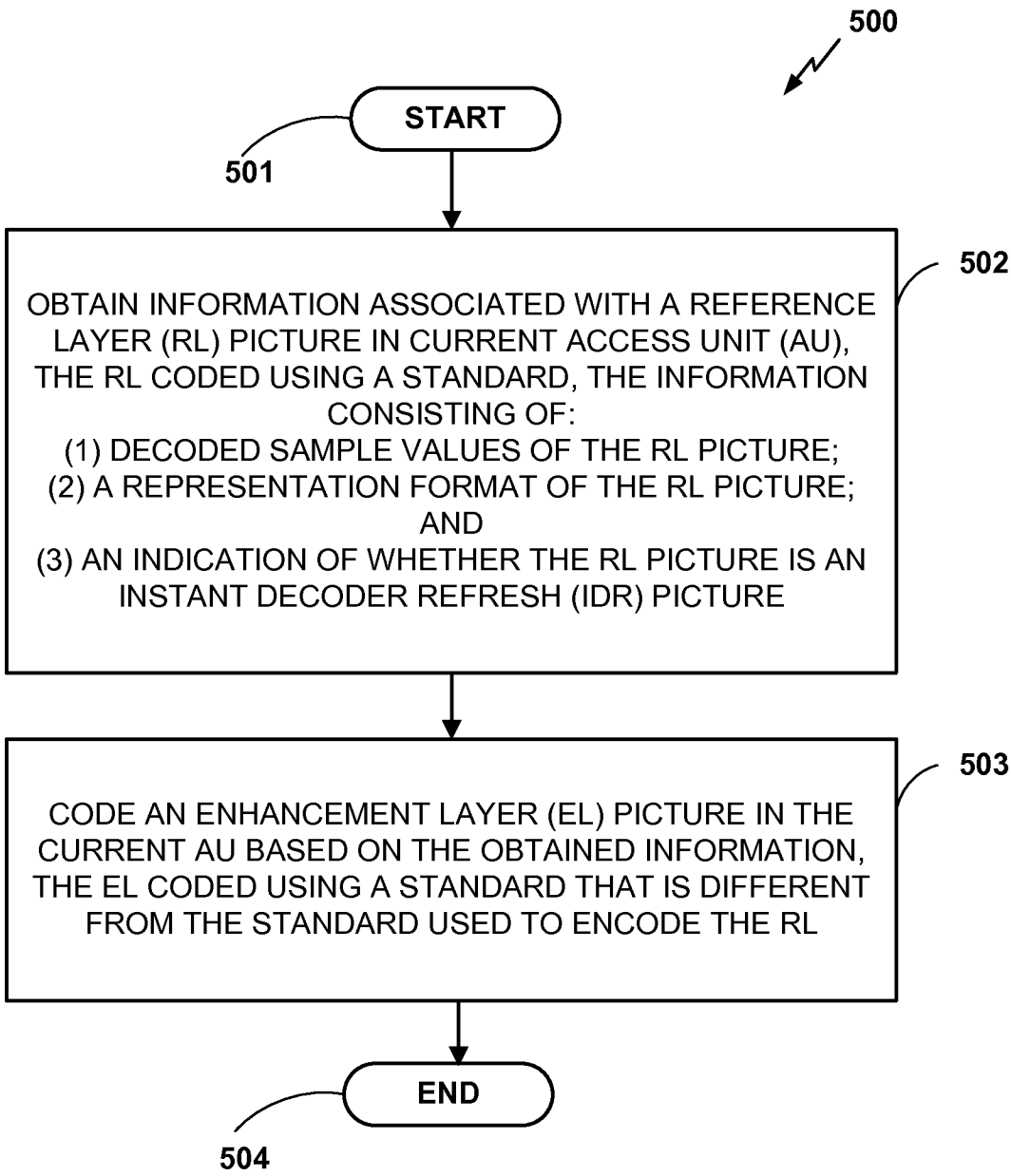
500
START
501
OBTAIN INFORMATION ASSOCIATED WITH A REFERENCE LAYER (RL) PICTURE IN CURRENT ACCESS UNIT (AU), THE RL CODED USING A STANDARD, THE INFORMATION CONSISTING OF:
(1) DECODED SAMPLE VALUES OF THE RL PICTURE;
(2) A REPRESENTATION FORMAT OF THE RL PICTURE; AND
(3) AN INDICATION OF WHETHER THE RL PICTURE IS AN INSTANT DECODER REFRESH (IDR) PICTURE
502
CODE AN ENHANCEMENT LAYER (EL) PICTURE IN THE CURRENT AU BASED ON THE OBTAINED INFORMATION, THE EL CODED USING A STANDARD THAT IS DIFFERENT FROM THE STANDARD USED TO ENCODE THE RL
503
END
504

FIG. 5

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2014/072698

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04N19/33 H04N19/70
ADD.

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

B. FIELDS SEARCHED

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

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

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X,P	Y-K WANG ET AL: "Support of AVC base layer in SHVC", 16. JCT-VC MEETING; 9-1-2014 - 17-1-2014; SAN JOSE; (JOINT COLLABORATIVE TEAM ON VIDEO CODING OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16); URL: HTTP://WFTP3.ITU.INT/AV-ARCH/JCTVC-SITE/,, no. JCTVC-P0184-v4, 13 February 2014 (2014-02-13), XP030115713, the whole document -----	1-30
X	EP 2 667 610 A2 (DOLBY LAB LICENSING CORP [US]) 27 November 2013 (2013-11-27) abstract; figures 3A, 3B, 5 paragraph [0003] paragraph [0045] - paragraph [0052] paragraph [0072] - paragraph [0074] ----- -/--	1-30

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

[X] See patent family annex.

* Special categories of cited documents :

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
26 March 2015	01/04/2015

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

2

Form PCT/ISA/210 (second sheet) (April 2005)

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	RAPAKA K ET AL: "MV-HEVC/SHVC HLS :Comments on latest MV-HEVC and SHVC draft specs", 15. JCT-VC MEETING; 23-10-2013 - 1-11-2013; GENEVA; (JOINT COLLABORATIVE TEAM ON VIDEO CODING OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16); URL: HTTP://WFTP3.ITU.INT/AV-ARCH/JCTVC-SITE/,, no. JCTVC-O0223, 16 October 2013 (2013-10-16), XP030115274, the whole document ----	1-30
A	LU S ET AL: "AHG9: Inter-layer RPS Prediction", 13. JCT-VC MEETING; 104. MPEG MEETING; 18-4-2013 - 26-4-2013; INCHEON; (JOINT COLLABORATIVE TEAM ON VIDEO CODING OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16); URL: HTTP://WFTP3.ITU.INT/AV-ARCH/JCTVC-SITE/,, no. JCTVC-M0140, 9 April 2013 (2013-04-09) , XP030114097, abstract Section 2 ----	1-30
A	NARASIMHAN S ET AL: "Consideration of buffer management issues and layer management in HEVC scalability", 14. JCT-VC MEETING; 25-7-2013 - 2-8-2013; VIENNA; (JOINT COLLABORATIVE TEAM ON VIDEO CODING OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16); URL: HTTP://WFTP3.ITU.INT/AV-ARCH/JCTVC-SITE/,, no. JCTVC-N0049, 12 July 2013 (2013-07-12) , XP030114477, abstract Sections 1.2, 2 ----	1-30

2

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2014/072698

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 2667610 A2	27-11-2013	EP 2667610 A2	27-11-2013
		US 2013314495 A1	28-11-2013