(12) STANDARD PATENT (11) Application No. AU 2012253847 B2
(19) AUSTRALIAN PATENT OFFICE

(54) Title
Offset type and coefficients signaling method for sample adaptive offset

(51) International Patent Classification(s)
H04N 19/61 (2014.01)

(21) Application No: **2012253847** (22) Date of Filing: **2012.05.04**

(87) WIPO No: **WO12/154576**

(30) Priority Data

(31) Number	(32) Date	(33) Country
61/484,624	**2011.05.10**	**US**
13/295,776	**2011.11.14**	**US**
61/503,454	**2011.06.30**	**US**

(43) Publication Date: **2012.11.15**
(44) Accepted Journal Date: **2015.12.03**

(71) Applicant(s)
Qualcomm Incorporated

(72) Inventor(s)
Chong, In Suk;Van Der Auwera, Geert;Karczewicz, Marta

(74) Agent / Attorney
Madderns Patent & Trade Mark Attorneys, GPO Box 2752, ADELAIDE, SA, 5001

(56) Related Art
EP 2237557

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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
15 November 2012 (15.11.2012)

WIPO | PCT

(10) International Publication Number
WO 2012/154576 A3

(51) International Patent Classification:
H04N 7/26 (2006.01)

(21) International Application Number:
PCT/US2012/036562

(22) International Filing Date:
4 May 2012 (04.05.2012)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/484,624 10 May 2011 (10.05.2011) US
61/503,454 30 June 2011 (30.06.2011) US
13/295,776 14 November 2011 (14.11.2011) US

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

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **CHONG, In Suk** [KR/US]; 5775 Morehouse Drive, San Diego, California 92121-1714 (US). **VAN DER AUWERA, Geert** [BE/US]; 5775 Morehouse Drive, San Diego, California 92121-1714 (US). **KARCZEWICZ, Marta** [US/US]; 5775 Morehouse Drive, San Diego, California 92121-1714 (US).

(74) Agent: **FITZGERALD, Kelly Patrick**; 1625 Radio Drive, Suite 300, Woodbury, Minnesota 55125 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, 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, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ,

[Continued on next page]

(54) Title: OFFSET TYPE AND COEFFICIENTS SIGNALING METHOD FOR SAMPLE ADAPTIVE OFFSET

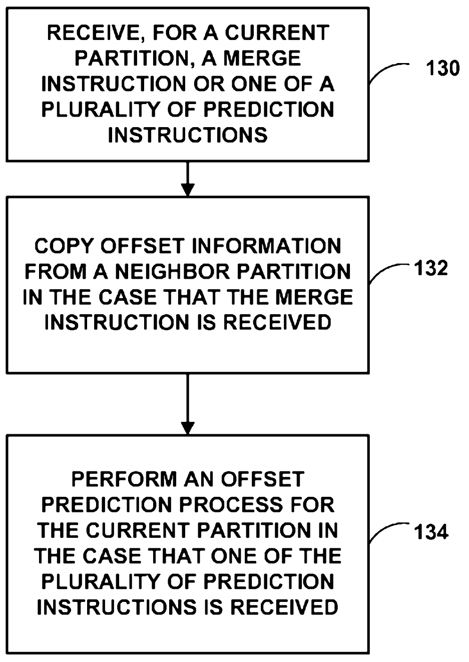

RECEIVE, FOR A CURRENT PARTITION, A MERGE INSTRUCTION OR ONE OF A PLURALITY OF PREDICTION INSTRUCTIONS
130
COPY OFFSET INFORMATION FROM A NEIGHBOR PARTITION IN THE CASE THAT THE MERGE INSTRUCTION IS RECEIVED
132
PERFORM AN OFFSET PREDICTION PROCESS FOR THE CURRENT PARTITION IN THE CASE THAT ONE OF THE PLURALITY OF PREDICTION INSTRUCTIONS IS RECEIVED
134

FIG. 10

(57) **Abstract**: This disclosure describes techniques for performing sample adaptive offset signaling and coding in a video coding process. Techniques of the disclosure include both a merge-based and prediction-based signaling process for sample adaptive offset information (i.e., offset values and offset type). The techniques includes determining offset information for a current partition, comparing the offset information of the current partition with offset information of one or more neighbor partitions, coding a merge instruction in the case that the offset information of one of the one or more neighbor partitions is the same as the offset information of the current partition, and coding one of a plurality of prediction instructions in the case that the offset information of the one or more neighbor partitions is not the same as the offset information of the current partition.

WO 2012/154576 A3

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, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

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

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

Published:

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

(88) Date of publication of the international search report: 2 May 2013

OFFSET TYPE AND COEFFICIENTS SIGNALING METHOD FOR SAMPLE ADAPTIVE OFFSET

[**0001**] This application claims the benefit of U.S. Provisional Application No. **61/484,624,** filed **May 10, 2011,** and U.S. Provisional Application No. **61/503,454,** filed **June 30, 2011,** both of which are hereby incorporated by reference in their entirety.

TECHNICAL FIELD

[**0002**] This disclosure relates to video coding, and more particularly to techniques for performing sample adaptive offset signaling and coding in a video coding process.

BACKGROUND

[**0003**] Digital video capabilities can be incorporated into a wide range of devices, including digital televisions, digital direct broadcast systems, wireless broadcast systems, personal digital assistants (PDAs), laptop or desktop computers, 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, to transmit, receive and store digital video information more efficiently.

[**0004**] Video compression techniques include spatial prediction and/or temporal prediction to reduce or remove redundancy inherent in video sequences. For block-based video coding, a video frame or slice may be partitioned into blocks. Each block can be further partitioned. Blocks in an intra-coded (I) frame or slice are encoded using spatial prediction with respect to reference samples in neighboring blocks in the same frame or slice. Blocks in an inter-coded (P or B) frame or slice may use spatial prediction with respect to reference samples in neighboring blocks in the same frame or slice or temporal prediction with respect to reference samples in other reference frames. Spatial or temporal prediction results in a predictive block for a block to be coded.

2012253847 11 Nov 2015

Residual data represents pixel differences between the original block to be coded and the predictive block.

[0005] 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 a particular order to produce a one-dimensional vector of transform coefficients for entropy coding.

SUMMARY

[0006] In general, this disclosure describes techniques for coding video data. This disclosure describes techniques for performing sample adaptive offset signaling and coding in a video coding process. Techniques of the disclosure include both a merge-based and prediction-based signaling process for sample adaptive offset information (i.e., offset values and offset type).

[0007] In one example of the disclosure, a method for encoding sample adaptive offset values in a video encoding process is proposed. The method includes,

determining sample adaptive offset information for a current partition;

performing a sample adaptive offset process for the current partition using the determined sample adaptive offset information;

comparing the sample offset information of the current partition with offset information of one or more neighbor partitions;

encoding a merge syntax element in the case that the sample adaptive offset information of one of the one or more neighbor partitions is the same as the sample adaptive offset information of the current partition, the merge syntax element indicating that a decoder is to copy the sample adaptive offset information for the current partition from the sample adaptive offset information of the one of the one or more neighbor partitions.

[0008] In another example of the disclosure, an apparatus configured to decode sample adaptive offset values in a video decoding process is proposed. The apparatus includes,

a memory configured to store data from an encoded video bitstream; and a processor in communication with the memory and configured to:

obtain from an encoded video bitstream, for a current partition, a merge syntax element indicating that sample adaptive offset information is to be copied from a specified neighbor partition;

copy sample adaptive offset information for the current partition from the specified neighbor partition in the case that the merge syntax element is obtained; and

perform a sample adaptive offset process for the current partition using the copied sample adaptive offset information.

[0009] In another example of the disclosure, a method for decoding sample adaptive offset values in a video decoding process is proposed. The method includes,

obtaining from an encoded video bitstream, for a current partition, a merge syntax element indicating that sample adaptive offset information is to be copied from a specified neighbor partition;

copying sample adaptive offset information for the current partition from the specified neighbour partition in the case that the merge syntax element is obtained; and

performing a sample adaptive offset prediction process for the current partition using the copied sample adaptive offset information.

Paragraph [0010] has been intentionally deleted.

[0011] The details of one or more examples are set forth in the accompanying drawings and the description below. Other features, objects, and advantages will be apparent from the description and drawings, and from the claims.

BRIEF DESCRIPTION OF DRAWINGS

[0012] FIG. 1 is a conceptual diagram depicting a quadtree structure for coding units in a video coding process.

[0013] FIG. 2 is a conceptual diagram showing the four possible edge offset types.

[0014] FIG. 3 is a conceptual diagram showing example band offset types.

[0015] FIG. 4 is a block diagram illustrating an example video encoding and decoding system.

[0016] FIG. 5 is a block diagram illustrating an example video encoder.

[0017] FIG. 6 is a conceptual diagram depicting potential neighbor partitions used in signaling offset information.

[0018] FIG. 7A is a conceptual diagram of an example of adaptive offset syntax coding in a merge mode.

[0019] FIG. 7B is a conceptual diagram of an example of adaptive offset syntax coding in a first prediction mode.

[0020] FIG. 7C is a conceptual diagram of an example of adaptive offset syntax coding in a second prediction mode.

[0021] FIG. 7D is a conceptual diagram of an example of adaptive offset syntax coding in a third prediction mode.

[0022] FIG. 8 is a block diagram illustrating an example video decoder.

[0023] FIG. 9 is a flowchart illustrating an example method of encoding video using sample adaptive offset.

[0024] FIG. 10 is a flowchart illustrating an example method of decoding video using sample adaptive offset.

DETAILED DESCRIPTION

[0025] In general, this disclosure describes techniques for coding video data. This disclosure describes techniques for coding sample adaptive offset values in a video coding process.

[0026] Digital video devices implement video compression techniques to encode and decode digital video information more efficiently. Video compression may apply spatial (intra-frame) prediction and/or temporal (inter-frame) prediction techniques to reduce or remove redundancy inherent in video sequences.

[0027] For video coding according to the high efficiency video coding (HEVC) standard currently under development by the Joint Cooperative Team for Video Coding (JCT-VC), as one example, a video frame may be partitioned into coding units. A coding unit (CU) generally refers to an image region that serves as a basic unit to which various coding tools are applied for video compression. A CU usually has a luminance component, denoted as Y, and two chroma components, denoted as U and V. Depending on the video sampling format, the size of the U and V components, in terms of number of samples, may be the same as or different from the size of the Y component. A CU is typically square, and may be considered to be similar to a so-called macroblock, e.g., under other video coding standards such as ITU-T H.264. Coding according to some of the presently proposed aspects of the developing HEVC standard will be described in this application for purposes of illustration. However, the techniques described in this disclosure may be useful for other video coding processes, such as those defined according to H.264 or other standard or proprietary video coding processes.

[0028] HEVC standardization efforts are based on a model of a video coding device referred to as the HEVC Test Model (HM). The HM presumes several capabilities of video coding devices over devices according to, e.g., ITU-T H.264/AVC. For example, whereas H.264 provides nine intra-prediction encoding modes, HM provides as many as thirty-four intra-prediction encoding modes.

[0029] According to the HM, a CU may include one or more prediction units (PUs) and/or one or more transform units (TUs). Syntax data within a bitstream may define a largest coding unit (LCU), which is a largest CU in terms of the number of pixels. In general, a CU has a similar purpose to a macroblock of H.264, except that a CU does not have a size distinction. Thus, a CU may be split into sub-CUs. In general, references in this disclosure to a CU may refer to a largest coding unit of a picture or a sub-CU of an LCU. An LCU may be split into sub-CUs, and each sub-CU may be further split into sub-CUs. Syntax data for a bitstream may define a maximum number of times an LCU may be split, referred to as CU depth. Accordingly, a bitstream may also define a smallest coding unit (SCU). This disclosure also uses the term "block" or "portion" to refer to any of a CU, PU, or TU. In general, "portion" may refer to any sub-set of a video frame.

[0030] An LCU may be associated with a quadtree data structure. FIG. 1 is a conceptual diagram of a quadtree splitting. In general, a quadtree data structure includes one node per CU, where a root node corresponds to the LCU. If a CU is split into four sub-CUs, the node corresponding to the CU includes four leaf nodes, each of which corresponds to one of the sub-CUs. Each node of the quadtree data structure may provide syntax data for the corresponding CU. For example, a node in the quadtree may include a split flag, indicating whether the CU corresponding to the node is split into sub-CUs. Syntax elements for a CU may be defined recursively, and may depend on whether the CU is split into sub-CUs. If a CU is not split further, it is referred as a leaf-CU. In this disclosure, 4 sub-CUs of a leaf-CU will also be referred to as leaf-CUs although there is no explicit splitting of the original leaf-CU. For example if a CU at

16x16 size is not split further, the four 8x8 sub-CUs will also be referred to as leaf-CUs although the 16x16 CU was never split.

[0031] Moreover, TUs of leaf-CUs may also be associated with respective quadtree data structures. That is, a leaf-CU may include a quadtree indicating how the leaf-CU is partitioned into TUs. This disclosure refers to the quadtree indicating how an LCU is partitioned as a CU quadtree and the quadtree indicating how a leaf-CU is partitioned into TUs as a TU quadtree. The root node of a TU quadtree generally corresponds to a leaf-CU, while the root node of a CU quadtree generally corresponds to an LCU. TUs of the TU quadtree that are not split are referred to as leaf-TUs.

[0032] A leaf-CU may include one or more prediction units (PUs). In general, a PU represents all or a portion of the corresponding CU, and may include data for retrieving a reference sample for the PU. For example, when the PU is inter-mode encoded, the PU may include data defining a motion vector for the PU. The data defining the motion vector may describe, for example, a horizontal component of the motion vector, a vertical component of the motion vector, a resolution for the motion vector (e.g., one-quarter pixel precision or one-eighth pixel precision), a reference frame to which the motion vector points, and/or a reference list (e.g., list 0 or list 1) for the motion vector. Data for the leaf-CU defining the PU(s) may also describe, for example, partitioning of the CU into one or more PUs. Partitioning modes may differ depending on whether the CU is uncoded, intra-prediction mode encoded, or inter-prediction mode encoded. For intra coding, a PU may be treated the same as a leaf transform unit described below.

[0033] Sample adaptive offset (SAO) coding is currently under consideration for adoption into the HEVC standard. In general, the addition of offset values to pixels in a video frame (e.g., a predictive frame for inter-prediction or intra-prediction) may improve coding during illumination changes between frames of a video sequence, e.g., such as during flashes, a darkening sky, or other types of illumination changes between frames. For example, offset values may be applied to pixels of a predicted video block in order to bias the values of the predictive video block so as to compensate for illumination changes. Previous video coding standards, such as H.264, applied offset types and values uniformly across entire blocks or frames of pixels. SAO techniques allow for different offset values to be applied to different pixels (or blocks) depending on pixel (or block) classification metrics. Possible classification metrics include activity metrics such as edge metrics and band metrics. A description of offset classifications can be found in C.-M. Fu, C.-Y. Chen, C.-Y. Tsai, Y.-W. Huang, S. Lei, "CE13: Sample Adaptive Offset with LCU-Independent Decoding", JCT-VC Contribution, E049, Geneva, Feb 2011.

[0034] In the current SAO implementation in the working draft of the HEVC standard, each partition (which consists of a set of LCUs) can have one of three offset types (also called pixel classification): no offset, band classification based offset type 0/1, and edge classification based type 0/1/2/3. Each band classification offset type has 16 possible offset values, while each edge classification based type has 4 possible offset values. If one of these offset types is chosen to be used for the partition, information indicating the corresponding offset type and the offset values are sent to the decoder.

[0035] The edge offset type classifies each pixel based on edge information. FIG. 2 is a conceptual diagram showing the four possible edge offset classifications (also called offset types) currently proposed for HEVC. An edge for a current pixel C is determined relative to two neighboring pixels (1 and 2). Edge offset type 0 (SAO_EO_0) is a horizontal edge. Edge offset type 1 (SAO_EO_1) is a vertical edge. Edge offset types 2 (SAO_EO_2) and 3 (SAO_EO_3) are diagonal edges. For each edge offset type value, four offset values are determined and signaled in the encoded video bitstream for use by a decoder.

[0036] For band offset, pixels are classified into different bands based on intensity. FIG. 3 is a conceptual diagram showing example bands based on intensity value. For band offset classification, pixels are categorized into 32 bands. The16 bands in the center are classified into one group (band offset type 0) and the remaining bands are classified into a second group (band offset type 1). For each group of bands, 16 offset values are determined and are signaled in the encoded video bitstream for use by a video decoder.

[0037] This disclosure proposes methods and techniques to more efficiently send offset type and offset values for each partition. In this context, a partition may include multiple coding units (CUs) (similar to so-called macroblocks in the H.264 standard) or even multiple largest coding units (LCUs).

[0038] In one example of the disclosure, techniques for encoding sample adaptive offset values in a video encoding process are proposed. The proposed techniques include determining offset information for a current partition, comparing the offset information of the current partition with offset information of one or more neighbor partitions, coding a merge instruction in the case that the offset information of one of the one or more neighbor partitions is the same as the offset information of the current partition, and coding one of a plurality of prediction instructions in the case that the offset information of the one or more neighbor partitions is not the same as the offset information of the current partition.

[0039] In another example of the disclosure, techniques for decoding sample adaptive offset values in a video decoding process are proposed. The proposed techniques include receiving, for a current partition, a merge instruction or one of a plurality of prediction instructions, copying offset information from a neighbor partition to the current partition in the case that the merge instruction is received, and performing an offset prediction process for the current partition in the case that one of the plurality of prediction instructions is received.

[0040] FIG. 4 is a block diagram illustrating an example video encoding and decoding system 10 that may be configured to utilize techniques for performing sample adaptive offset signaling and coding in accordance with examples of this disclosure. As shown in FIG. 4, the system 10 includes a source device 12 that transmits encoded video to a destination device 14 via a communication channel 16. Encoded video data may also be stored on a storage medium 34 or a file server 36 and may be accessed by the destination device 14 as desired. When stored to a storage medium or file server, video encoder 20 may provide coded video data to another device, such as a network interface, a compact disc (CD), Blu-ray or digital video disc (DVD) burner or stamping facility device, or other devices, for storing the coded video data to the storage medium. Likewise, a device separate from video decoder 30, such as a network interface, CD or DVD reader, or the like, may retrieve coded video data from a storage medium and provided the retrieved data to video decoder 30.

[0041] The source device 12 and the destination device 14 may comprise any of a wide variety of devices, including desktop computers, notebook (i.e., laptop) computers, tablet computers, set-top boxes, telephone handsets such as so-called smartphones, televisions, cameras, display devices, digital media players, video gaming consoles, or the like. In many cases, such devices may be equipped for wireless communication. Hence, the communication channel 16 may comprise a wireless channel, a wired channel, or a combination of wireless and wired channels suitable for transmission of encoded video data. Similarly, the file server 36 may be accessed by the destination device 14 through any standard data connection, including an Internet connection. This may include a wireless channel (e.g., a Wi-Fi connection), a wired connection (e.g., DSL, cable modem, etc.), or a combination of both that is suitable for accessing encoded video data stored on a file server.

[0042] Techniques for performing sample adaptive offset signaling and coding, in accordance with examples of this disclosure, may be applied to video coding in support of any of a variety of multimedia applications, such as over-the-air television broadcasts, cable television transmissions, satellite television transmissions, streaming video transmissions, e.g., via the Internet, 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, the 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.

[0043] In the example of FIG. 4, the source device 12 includes a video source 18, a video encoder 20, a modulator/demodulator 22 and a transmitter 24. In the source device 12, the video source 18 may include a source such as a video capture device, such as 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. 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, or application in which encoded video data is stored on a local disk.

[0044] The captured, pre-captured, or computer-generated video may be encoded by the video encoder 20. The encoded video information may be modulated by the modem 22 according to a communication standard, such as a wireless communication protocol, and transmitted to the destination device 14 via the transmitter 24. The modem 22 may include various mixers, filters, amplifiers or other components designed for signal modulation. The transmitter 24 may include circuits designed for transmitting data, including amplifiers, filters, and one or more antennas.

[0045] The captured, pre-captured, or computer-generated video that is encoded by the video encoder 20 may also be stored onto a storage medium 34 or a file server 36 for later consumption. The storage medium 34 may include Blu-ray discs, DVDs, CD-ROMs, flash memory, or any other suitable digital storage media for storing encoded video. The encoded video stored on the storage medium 34 may then be accessed by the destination device 14 for decoding and playback.

[0046] The file server 36 may be any type of server capable of storing encoded video and transmitting that encoded video to the destination device 14. Example file servers include a web server (e.g., for a website), an FTP server, network attached storage (NAS) devices, a local disk drive, or any other type of device capable of storing encoded video data and transmitting it to a destination device. The transmission of encoded video data from the file server 36 may be a streaming transmission, a download transmission, or a combination of both. The file server 36 may be accessed by the destination device 14 through any standard data connection, including an Internet connection. This may include a wireless channel (e.g., a Wi-Fi connection), a wired connection (e.g., DSL, cable modem, Ethernet, USB, etc.), or a combination of both that is suitable for accessing encoded video data stored on a file server.

[0047] The destination device 14, in the example of FIG. 4, includes a receiver 26, a modem 28, a video decoder 30, and a display device 32. The receiver 26 of the destination device 14 receives information over the channel 16, and the modem 28 demodulates the information to produce a demodulated bitstream for the video decoder 30. The information communicated over the channel 16 may include a variety of syntax information generated by the video encoder 20 for use by the video decoder 30 in decoding video data. Such syntax may also be included with the encoded video data stored on the storage medium 34 or the file server 36. Each of the video encoder 20 and the video decoder 30 may form part of a respective encoder-decoder (CODEC) that is capable of encoding or decoding video data. In this disclosure, the term coder refers to an encoder, a decoder, or CODEC, and the terms coder, encoder, decoder and CODEC all refer to specific machines designed for the coding (encoding and/or decoding) of video data consistent with this disclosure.

[0048] 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.

[0049] In the example of FIG. 4, the communication channel 16 may comprise any wireless or wired communication medium, such as a radio frequency (RF) spectrum or one or more physical transmission lines, or any combination of wireless and wired media. The communication channel 16 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 channel 16 generally represents any suitable communication medium, or collection of different communication media, for transmitting video data from the source device 12 to the destination device 14, including any suitable combination of wired or wireless media. The communication channel 16 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.

[0050] The video encoder 20 and the video decoder 30 may operate according to a video compression standard, such as the High Efficiency Video Coding (HEVC) standard presently under development, and may conform to the HEVC Test Model (HM). Alternatively, the video encoder 20 and the video decoder 30 may operate according to other proprietary or industry standards, such as the ITU-T H.264 standard, alternatively referred to as MPEG-4, Part 10, Advanced Video Coding (AVC), or extensions of such standards. The techniques of this disclosure, however, are not limited to any particular coding standard. Other examples include MPEG-2 and ITU-T H.263.

[0051] Although not shown in FIG. 4, in some aspects, 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).

[0052] 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.

[0053] The video encoder 20 may implement any or all of the techniques of this disclosure for performing sample adaptive offset signaling and coding in a video encoding process. Likewise, the video decoder 30 may implement any or all of these techniques for performing sample adaptive offset coding in a video coding process. A video coder, as described in this disclosure, may refer to a video encoder or a video decoder. Similarly, a video coding unit may refer to a video encoder or a video decoder. Likewise, video coding may refer to video encoding or video decoding.

[0054] In one example of the disclosure, the video encoder 20 of the source device 12 may be configured to compare offset information of a current partition with offset information of one or more neighbor partitions, code a merge instruction in the case that the offset information of one of the one or more neighbor partitions is the same as the offset information of the current partition, and code one of a plurality of prediction instructions in the case that the offset information of the one or more neighbor partitions is not the same as the offset information of the current partition.

[0055] In another example of the disclosure, the video decoder 30 of the destination device 14 may be configured to receive, for a current partition, a merge instruction or one of a plurality of prediction instructions, copy offset information from a neighbor partition to the current partition in the case that the merge instruction is received, perform an offset prediction process for the current partition in the case that one of the plurality of prediction instructions is received.

[0056] FIG. 5 is a block diagram illustrating an example of a video encoder 20 that may use techniques for coding and signaling sample adaptive offset values as described in this disclosure. The video encoder 20 will be described in the context of HEVC coding for purposes of illustration, but without limitation of this disclosure as to other coding standards or methods that may require scanning of transform coefficients. The video encoder 20 may perform intra- and inter-coding of CUs within video frames. Intra-coding relies on spatial prediction to reduce or remove spatial redundancy in video data within a given video frame. Inter-coding relies on temporal prediction to reduce or remove temporal redundancy between a current frame and previously coded frames of a video sequence. Intra-mode (I-mode) may refer to any of several spatial-based video compression modes. Inter-modes such as uni-directional prediction (P-mode) or bi-directional prediction (B-mode) may refer to any of several temporal-based video compression modes.

[0057] As shown in FIG. 5, the video encoder 20 receives a current video block within a video frame to be encoded. In the example of FIG. 5, the video encoder 20 includes a motion compensation unit 44, a motion estimation unit 42, an intra-prediction unit 46, a reference frame buffer 64, a summer 50, a transform unit 52, a quantization unit 54, and an entropy encoding unit 56. The transform unit 52 illustrated in FIG. 5 is the unit that applies the actual transform or combinations of transform to a block of residual data, and is not to be confused with block of transform coefficients, which also may be referred to as a transform unit (TU) of a CU. For video block reconstruction, the video encoder 20 also includes an inverse quantization unit 58, an inverse transform unit 60, a summer 62, and an SAO unit 43. A deblocking filter may also be included to filter block boundaries to remove blockiness artifacts from reconstructed video. If desired, the deblocking filter would typically filter the output of the summer 62.

[0058] During the encoding process, the video encoder 20 receives a video frame or slice to be coded. The frame or slice may be divided into multiple video blocks, e.g., largest coding units (LCUs). The motion estimation unit 42 and the motion compensation unit 44 perform inter-predictive coding of the received video block relative to one or more blocks in one or more reference frames to provide temporal compression. The intra-prediction unit 46 may perform intra-predictive coding of the received video block relative to one or more neighboring blocks in the same frame or slice as the block to be coded to provide spatial compression.

[0059] The mode select unit 40 may select one of the coding modes, intra or inter, e.g., based on rate distortion results for each mode, and provides the resulting intra- or inter-predicted block (e.g., a prediction unit (PU)) to the summer 50 to generate residual block data and to the summer 62 to reconstruct the encoded block for use in a reference frame. Summer 62 combines the predicted block with inverse quantized, inverse transformed data from inverse transform unit 60 for the block to reconstruct the encoded block, as described in greater detail below. Some video frames may be designated as I-frames, where all blocks in an I-frame are encoded in an intra-prediction mode. In some cases, the intra-prediction unit 46 may perform intra-prediction encoding of a block in a P- or B-frame, e.g., when motion search performed by the motion estimation unit 42 does not result in a sufficient prediction of the block.

[0060] The motion estimation unit 42 and the motion compensation unit 44 may be highly integrated, but are illustrated separately for conceptual purposes. Motion estimation (or motion search) is the process of generating motion vectors, which estimate motion for video blocks. A motion vector, for example, may indicate the displacement of a prediction unit in a current frame relative to a reference sample of a reference frame. The motion estimation unit 42 calculates a motion vector for a prediction unit of an inter-coded frame by comparing the prediction unit to reference samples of a reference frame stored in the reference frame buffer 64. A reference sample may be a block that is found to closely match the portion of the CU including the PU being coded in terms of pixel difference, which may be determined by sum of absolute difference (SAD), sum of squared difference (SSD), or other difference metrics. The reference sample may occur anywhere within a reference frame or reference slice, and not necessarily at a block (e.g., coding unit) boundary of the reference frame or slice. In some examples, the reference sample may occur at a fractional pixel position.

[0061] The motion estimation unit 42 sends the calculated motion vector to the entropy encoding unit 56 and the motion compensation unit 44. The portion of the reference frame identified by a motion vector may be referred to as a reference sample. The motion compensation unit 44 may calculate a prediction value for a prediction unit of a current CU, e.g., by retrieving the reference sample identified by a motion vector for the PU.

[0062] The intra-prediction unit 46 may intra-predict the received block, as an alternative to inter-prediction performed by the motion estimation unit 42 and the motion compensation unit 44. The intra-prediction unit 46 may predict the received block relative to neighboring, previously coded blocks, e.g., blocks above, above and to the right, above and to the left, or to the left of the current block, assuming a left-to-right, top-to-bottom encoding order for blocks. The intra-prediction unit 46 may be configured with a variety of different intra-prediction modes. For example, the intra-prediction unit 46 may be configured with a certain number of directional prediction modes, e.g., thirty-five directional prediction modes, based on the size of the CU being encoded.

[0063] The intra-prediction unit 46 may select an intra-prediction mode by, for example, calculating error values for various intra-prediction modes and selecting a mode that yields the lowest error value. Directional prediction modes may include functions for combining values of spatially neighboring pixels and applying the combined values to one or more pixel positions in a PU. Once values for all pixel positions in the PU have been calculated, the intra-prediction unit 46 may calculate an error value for the prediction mode based on pixel differences between the PU and the received block to be encoded. The intra-prediction unit 46 may continue testing intra-prediction modes until an intra-prediction mode that yields an acceptable error value is discovered. The intra-prediction unit 46 may then send the PU to the summer 50.

[0064] The video encoder 20 forms a residual block by subtracting the prediction data calculated by the motion compensation unit 44 or the intra-prediction unit 46 from the original video block being coded. The summer 50 represents the component or components that perform this subtraction operation. The residual block may correspond to a two-dimensional matrix of pixel difference values, where the number of values in the residual block is the same as the number of pixels in the PU corresponding to the residual block. The values in the residual block may correspond to the differences, i.e., error, between values of co-located pixels in the PU and in the original block to be coded. The differences may be chroma or luma differences depending on the type of block that is coded.

[0065] The transform unit 52 may form one or more transform units (TUs) from the residual block. The transform unit 52 selects a transform from among a plurality of transforms. The transform may be selected based on one or more coding characteristics, such as block size, coding mode, or the like. The transform unit 52 then applies the selected transform to the TU, producing a video block comprising a two-dimensional array of transform coefficients. The transform unit 52 may select the transform partition according to above-described techniques of this disclosure. In addition, the transform unit 52 may signal the selected transform partition in the encoded video bitstream.

[0066] The transform unit 52 may send the resulting transform coefficients to the quantization unit 54. The quantization unit 54 may then quantize the transform coefficients. The entropy encoding unit 56 may then perform a scan of the quantized transform coefficients in the matrix according to a scanning mode. This disclosure describes the entropy encoding unit 56 as performing the scan. However, it should be understood that, in other examples, other processing units, such as the quantization unit 54, could perform the scan.

[0067] Once the transform coefficients are scanned into the one-dimensional array, the entropy encoding unit 56 may apply entropy coding such as CAVLC, CABAC, syntax-based context-adaptive binary arithmetic coding (SBAC), or another entropy coding methodology to the coefficients.

[0068] To perform CAVLC, the entropy encoding unit 56 may select a variable length code for a symbol to be transmitted. Codewords in VLC may be constructed such that relatively shorter codes correspond to more likely symbols, while longer codes correspond to less likely symbols. In this way, the use of VLC may achieve a bit savings over, for example, using equal-length codewords for each symbol to be transmitted.

[0069] To perform CABAC, the entropy encoding unit 56 may select a context model to apply to a certain context to encode symbols to be transmitted. The context may relate to, for example, whether neighboring values are non-zero or not. The entropy encoding unit 56 may also entropy encode syntax elements, such as the signal representative of the selected transform. In accordance with the techniques of this disclosure, the entropy encoding unit 56 may select the context model used to encode these syntax elements based on, for example, an intra-prediction direction for intra-prediction modes, a scan position of the coefficient corresponding to the syntax elements, block type, and/or transform type, among other factors used for context model selection.

[0070] Following the entropy coding by the entropy encoding unit 56, the resulting encoded video may be transmitted to another device, such as the video decoder 30, or archived for later transmission or retrieval.

[0071] In some cases, the entropy encoding unit 56 or another unit of the video encoder 20 may be configured to perform other coding functions, in addition to entropy coding. For example, the entropy encoding unit 56 may be configured to determine coded block pattern (CBP) values for CU's and PU's. Also, in some cases, the entropy encoding unit 56 may perform run length coding of coefficients.

[0072] The inverse quantization unit 58 and the inverse transform unit 60 apply inverse quantization and inverse transformation, respectively, to reconstruct the residual block in the pixel domain, e.g., for later use as a reference block. The motion compensation unit 44 may calculate a reference block by adding the residual block to a predictive block of one of the frames of the reference frame buffer 64. The motion compensation unit 44 may also apply one or more interpolation filters to the reconstructed residual block to calculate sub-integer pixel values for use in motion estimation. The summer 62 adds the reconstructed residual block to the motion compensated prediction block produced by the motion compensation unit 44 to produce a reconstructed video block.

[0073] The SAO unit 43 determines offset type and offset values for a current partition and adds offset values to the pixels of the reconstructed video block (i.e., the current partition) and signals adaptive offset syntax in the encoded video bitstream for use by a decoder. The reconstructed video block with offset values is then stored in the reference frame buffer 64. The reconstructed video block may be used by the motion estimation unit 42 and the motion compensation unit 44 as a reference block to inter-code a block in a subsequent video frame.

[0074] In some cases, offset information (i.e., offset type and offset values) is signaled in the encoded video bitstream by sending them as they are, i.e., without prediction or reference to other partitions. However, this technique fails to make use of possible correlations between different partitions, which can be exploited in the coding of offset type and offset values for increased efficiency. This disclosure proposes the following techniques take advantage of the potential correlations to code offset type and values.

[0075] Initially, it is determined if the offset type and offset values are applied at the slice level or at the partition level. That is to say, the encoder may choose whether a whole slice has one offset type and offset value (slice based) or whether each partition has its own offset type and values (partition based). If slice based offset is chosen, offset type and offset values are sent to the decoder with no prediction, in which case the offset values may be sent in a slice-level syntax (e.g., the slice header) and used for CUs within that slice. Otherwise, if partition based offsets are selected, the encoder may employ merge-based or prediction-based signaling of offset information.

[0076] SAO unit 43 may signal a merge instruction in the encoded video bitstream that instructs a decoder to simply copy the offset type and values from a neighbor partition for the current partition. Alternatively, the SAO unit 43 may signal a prediction instruction in the encoded video bitstream that instructs a decode to apply a prediction technique to determine the offset type and offset values for the current partition (no merge).

[0077] Both merge-based and prediction-based signaling of offset information for a current partition may relay on the offset information of neighboring partitions relative to the current partition. FIG. 6 is a conceptual diagram depicting potential neighbor partitions used in signaling offset information. Neighbor partitions are usually derived in the causal fashion (i.e., neighbor partitions that have already been decoded by the decoder). Examples of locations of neighbor partition include the upper (U), left (L), upper left (UL), and upper right (UR) partitions relative to the current partition. Neighbor partitions may be directly adjacent the current partition. In another example, neighbor partitions may be further away from the current partition, but in the same coding unit (e.g., within the same LCU).

[0078] Neighbor partitions that are potential merge candidates are available (i.e., causal) neighbor partitions with the same offset type. For example, if the current partition's offset type is edge offset type 0 (EO0), causal neighbor partitions with offset type EO0 are considered to be merge candidates.

[0079] Different cases may apply when determining available neighbor partitions when quad-tree based partitioning is used. FIG. 6 shows a case 1 where the direct (i.e., adjacent) neighbor partition of the same size has the same offset type. In case 2, the parent of the neighbor has the same offset type (i.e., a partition of larger size than the current partition). In case3, the direct neighbor partition of the same size does not have the same offset type, but a neighbor partition of the direct neighbor partition does have the same offset type. Each of these three cases shows examples of neighbor partitions that may be used for the sample adaptive offset merge mode.

[0080] After determining the offset type and offset values for a current partition, the SAO unit 43 compares the offset type and offset values of the current partition to the neighbor partitions. Merge-mode is used when one of the neighbor partitions has the same offset type and offset values as the current partition.

[0081] FIG. 7A is a conceptual diagram of an example of adaptive offset syntax coding in a merge mode. A "merge" flag 1 (e.g., 1 for merge and 0 for no merge, or vice versa) is signaled in the encoded video bitstream to indicate whether the decoder should copy a neighboring partition's offset type and offset values for the current partition (i.e., perform a merge process). Note that the merge flag can be inferred to be 0 (i.e., no merge) when there is no available neighbor with offset enabled.

[0082] If merge is used (i.e., merge flag set to 1), in addition to the merge flag 1, the SAO unit 43 signals a neighbor partition index 3 to the decoder which indicates the index of the neighbor partition whose offset information (i.e., offset type and values) will be copied to the current partition during video decoding. The signaling of the neighbor partition index depends on the number of different available neighbor partitions (also called merge candidates). The neighbor partition index may not be sent when there is only one merge candidate. In that situation, the decoder may infer the index of the only merge candidate.

[0083] If the SAO unit 43 compares the offset information of the current partition to all of the neighbor partitions and does not find an exact match, merge is not used. If no merge is used, (i.e., merge flag set to 0), a prediction type is sent to the decoder (e.g., using a pred_type flag). This disclosure proposes three possible prediction types.

[0084] FIG. 7B is a conceptual diagram of an example of adaptive offset syntax coding in a first prediction mode. A first prediction instruction may be sent (i.e., pred_type 0) is the case that the SAO unit 43 determines that the offset values of a neighbor partition are not identical to those of the current partition, but are close enough so that bit rate efficiencies can be achieved by sending the residual offset values. The use of pred_type 0 may be determined by comparing the bit rate achieved by sending residual offset values as compared to the bit rate from sending the offset values of the current partition as is.

[0085] For the first prediction instruction, a pred_type flag 5 is set to 0. The neighbor partition index 7 is the index of the predictive neighbor partition. Like merge mode, pred_type 0 uses neighbor partitions that have the same offset type of the current partition. A decoder may simply copy the offset type for the current partition from the neighbor partition during the decoding process. Residual offset values 9 are also signaled in the encoded video bitstream. The residual offset values are created by subtracting the offset values determined for the current partition, from the offset values of the selected neighbor partition. A decoder may then simply add the received residual offset values to the offset values of the neighbor partition indicated by the neighbor partition index 7 to reconstruct the offset values for the current partition. Neighbor partitions that are available as prediction candidates may be derived in the same way as for merge mode described above with reference to FIG. 6. If there is only one available neighbor partition, neighbor partition index may not be signaled, but instead may be inferred by the decoder.

[0086] In situations where no neighbor partition has offset values close enough to those of the current partition to achieve coding efficiencies by sending the residual, a second prediction instruction (pred_type 1) can be signaled in the encoded video bitstream. FIG. 7C is a conceptual diagram of an example of adaptive offset syntax coding in a second prediction mode. In pred_type 1, both the offset type 11 and offset values 13 for the current partition are signaled in the encoded bitstream as is. Essentially, the decoder need not reference any neighbor partition in this mode, but may simply receive the offset type and offset values for the current partition.

[0087] A third prediction instruction (pred_type 2) may be sent in situations where the SAO unit 43 determines that the current partition's offset values are the same as a neighboring partition, but the offset type is different. FIG. 7D is a conceptual diagram of an example of adaptive offset syntax coding in a third prediction mode. In this case, the neighbor partition index 7 indicates the neighboring partition from which to copy offset values for the current partition. In addition, the offset type 11 for the current partition is also signaled in the encoded video bitstream. Note that this differs slightly from the merge flag, as only the offset values are copied from a neighbor partition and not the offset type. The derivation of prediction candidates for prediction type 2 is slightly different than that for prediction type 0 and merge mode. Instead of considering causal neighbors with the same offset type, prediction type 2 considers neighbor partitions with different offset types. For example, if the current partition's offset type is edge offset type 0 (EO0), neighbors with EO1/2/3 are considered to be prediction candidates.

[0088] FIG. 8 is a block diagram illustrating an example of a video decoder 30, which decodes an encoded video sequence. In the example of FIG. 8, the video decoder 30 includes an entropy decoding unit 70, a motion compensation unit 72, an intra-prediction unit 74, an inverse quantization unit 76, an inverse transformation unit 78, a reference frame buffer 82, a SAO unit 79, and a summer 80. The video decoder 30 may, in some examples, perform a decoding pass generally reciprocal to the encoding pass described with respect to the video encoder 20 (see FIG. 5).

[0089] The entropy decoding unit 70 performs an entropy decoding process on the encoded bitstream to retrieve a one-dimensional array of transform coefficients. The entropy decoding process used depends on the entropy coding used by the video encoder 20 (e.g., CABAC, CAVLC, etc.). The entropy coding process used by the encoder may be signaled in the encoded bitstream or may be a predetermined process.

[0090] In some examples, the entropy decoding unit 70 (or the inverse quantization unit 76) may scan the received values using a scan mirroring the scanning mode used by the entropy encoding unit 56 (or the quantization unit 54) of the video encoder 20. Although the scanning of coefficients may be performed in the inverse quantization unit 76, scanning will be described for purposes of illustration as being performed by the entropy decoding unit 70. In addition, although shown as separate functional units for ease of illustration, the structure and functionality of the entropy decoding unit 70, the inverse quantization unit 76, and other units of the video decoder 30 may be highly integrated with one another.

[0091] The inverse quantization unit 76 inverse quantizes, i.e., de-quantizes, the quantized transform coefficients provided in the bitstream and decoded by the entropy decoding unit 70. The inverse quantization process may include a conventional process, e.g., similar to the processes proposed for HEVC or defined by the H.264 decoding standard. The inverse quantization process may include use of a quantization parameter QP calculated by the video encoder 20 for the CU to determine a degree of quantization and, likewise, a degree of inverse quantization that should be applied. The inverse quantization unit 76 may inverse quantize the transform coefficients either before or after the coefficients are converted from a one-dimensional array to a two-dimensional array.

[0092] The inverse transform unit 78 applies an inverse transform to the inverse quantized transform coefficients. In some examples, the inverse transform unit 78 may determine an inverse transform based on signaling from the video encoder 20, or by inferring the transform from one or more coding characteristics such as block size, coding mode, or the like. In some examples, the inverse transform unit 78 may determine a transform to apply to the current block based on a signaled transform at the root node of a quadtree for an LCU including the current block. Alternatively, the transform may be signaled at the root of a TU quadtree for a leaf-node CU in the LCU quadtree. In some examples, the inverse transform unit 78 may apply a cascaded inverse transform, in which inverse transform unit 78 applies two or more inverse transforms to the transform coefficients of the current block being decoded.

[0093] The intra-prediction unit 74 may generate prediction data for a current block of a current frame based on a signaled intra-prediction mode and data from previously decoded blocks of the current frame.

[0094] Based on the retrieved motion prediction direction, reference frame index, and calculated current motion vector, the motion compensation unit produces a motion compensated block for the current portion. These motion compensated blocks essentially recreate the predictive block used to produce the residual data.

[0095] The motion compensation unit 72 may produce the motion compensated blocks, possibly performing interpolation based on interpolation filters. Identifiers for interpolation filters to be used for motion estimation with sub-pixel precision may be included in the syntax elements. The motion compensation unit 72 may use interpolation filters as used by the video encoder 20 during encoding of the video block to calculate interpolated values for sub-integer pixels of a reference block. The motion compensation unit 72 may determine the interpolation filters used by the video encoder 20 according to received syntax information and use the interpolation filters to produce predictive blocks.

[0096] Additionally, the motion compensation unit 72 and the intra-prediction unit 74, in an HEVC example, may use some of the syntax information (e.g., provided by a quadtree) to determine sizes of LCUs used to encode frame(s) of the encoded video sequence. The motion compensation unit 72 and the intra-prediction unit 74 may also use syntax information to determine split information that describes how each CU of a frame of the encoded video sequence is split (and likewise, how sub-CUs are split). The syntax information may also include modes indicating how each split is encoded (e.g., intra- or inter-prediction, and for intra-prediction an intra-prediction encoding mode), one or more reference frames (and/or reference lists containing identifiers for the reference frames) for each inter-encoded PU, and other information to decode the encoded video sequence.

[0097] The summer 80 combines the residual blocks with the corresponding prediction blocks generated by the motion compensation unit 72 or the intra-prediction unit 74 to form decoded blocks. The SAO unit 79 than removes the offset introduced by the encoder. The SAO unit 79 performs this process according to received syntax that indicates the offset information for a partition, such as the syntax described above with reference to FIGs. 7A-D.

[0098] If a merge instruction is received, the SAO unit 79 copies offset information to the current partition from the neighbor partition indicated by the neighbor partition index 3 (see FIG. 7A).

[0099] If a first prediction instruction (pred_type 0) is received, the SAO unit 79 copies the offset type to the current partition from the neighbor partition indicated by the neighbor partition index 7 (see FIG. 7B). In addition, the SAO unit 79 adds the received residual offset values 9 to the offset values of the neighbor partition index indicated by the neighbor partition index 7 to reconstruct offset values for the current partition (see FIG. 7B).

[0100] If a second prediction instruction (pred_type 1) is received, the SAO unit 79 uses the received offset type 11 and offset values 13 as the offset information for the current partition (see FIG. 7C).

[0101] If a third prediction instruction (pred_type 2) is received, the SAO unit 79 copies to the current partition the offset values from the neighbor partition indicated by the neighbor partition index 7 (see FIG. 7D). In addition, the SAO unit 79 uses the received offset type 11 as the offset type for the current partition.

[0102] If desired, a deblocking filter may also be applied to filter the decoded blocks in order to remove blockiness artifacts. Additional filtering may also be applied either in-loop or post-loop. The decoded video blocks are then stored in the reference frame buffer 82, which provides reference blocks for subsequent motion compensation and also produces decoded video for presentation on a display device (such as the display device 32 of FIG. 4). If in-loop filtering is applied, the decoded video blocks stored for subsequent motion compensation are filtered blocks, but if post-loop filtering is applied, the decoded video blocks stored for subsequent motion compensation are unfiltered blocks.

[0103] FIG. 9 is a flowchart illustrating an example method of encoding video using sample adaptive offset. The method of FIG. 9 is described from the perspective of video encoder 20 shown in FIG. 5, although other devices may perform the techniques of FIG. 9. As shown in FIG. 9, the video encoder 20 may be configured to encode sample adaptive offset values in a video encoding process. The video encoder 20 may determine offset information for a current partition (120) and compare the offset information of the current partition with offset information of one or more neighbor partitions (122). Offset information may include an offset type and offset values. The video encoder 20 codes a merge instruction in the case that the offset information of one of the one or more neighbor partitions is the same as the offset information of the current partition (124). The video encoder 20 codes one of a plurality of prediction instructions in the case that the offset information of the one or more neighbor partitions is not the same as the offset information of the current partition (124).

[0104] The merge instruction instructs a decoder to copy the offset information of one of the one or more neighbor partitions for the current partition. The merge instruction may include a merge flag and an index of the one of the one or more neighbor partitions with the same offset information as the current partition.

[0105] In one example, coding one of a plurality of prediction instructions may include coding a first prediction instruction which instructs a decoder to derive offset information for the current partition from offset information of one of the one or more neighbor partitions. The first prediction instruction may include an index of one of the one or more neighbor partitions and offset residual values, and wherein offset residual values are created by subtracting the offset values of the current partition from offset values of one of the one or more neighbor partitions. The video encoder 20 may code the first prediction in the case that bit rate efficiencies may be achieved by send offset residual values.

[0106] In another example, coding one of a plurality of prediction instructions includes coding a second prediction instruction which includes offset information for the current partition. The video encoder 20 may code the second prediction instruction in the case that no bit rate efficiencies may be achieved by sending offset residual values.

[0107] In another example, coding one of a plurality of prediction instructions includes coding a third prediction instruction which includes an index of one of the one or more neighbor partitions from which to copy offset values and also includes an offset type for the current partition.

[0108] FIG. 10 is a flowchart illustrating an example method of decoding video using sample adaptive offset. The method of FIG. 10 is described from the perspective of the video decoder 30 shown in FIG. 8, although other devices may perform the techniques of FIG. 10. As shown in FIG. 10, the video decoder 30 may be configured to decode sample adaptive offset values in a video decoding process. The video decoder may be configured to receive, for a current partition, a merge instruction or one of a plurality of prediction instructions (130). The video decoder is configured to copy offset information from a neighbor partition to the current partition in the case that the merge instruction is received (132). The merge instruction includes a merge flag and an index of the neighbor partition. The offset information may include an offset type and offset values.

[0109] The video decoder is also configured to perform an offset prediction process for the current partition in the case that one of the plurality of prediction instructions is received (134).

[0110] The plurality of prediction instructions may include a first, second, and third prediction instruction. In the case that the first prediction instruction is received, the offset prediction process derives offset information for the current partition from offset information of one of the one or more neighbor partitions. The first prediction instruction may include an index of one of the one or more neighbor partitions and offset residual values. The video decoder 20 derives offset values for the current partition by adding offset values of the one neighbor partition to the offset residual values.

[0111] In the case that the second prediction instruction is received, the video decoder 20 also receives the offset information. The received offset information is used for the current partition.

[0112] In the case that the third prediction instruction is received, the video decoder 20 also receives an index of one of the one or more neighbor partitions from which to copy offset values and also receives an offset type. The video decoder 20 performs the offset prediction process by deriving offset information for the current partition by copying offset values from the one neighbor partition for the current partition and by assigning the received offset type to the current partition.

[0113] In one or more examples, the functions described may be implemented in hardware, software, firmware, or any combination thereof. If implemented in software, the functions may be stored on or transmitted over, as one or more instructions or code, a computer-readable medium and executed by a hardware-based processing unit. Computer-readable media may include computer-readable storage media, which corresponds to a tangible medium such as data storage media, or communication media including any medium that facilitates transfer of a computer program from one place to another, e.g., according to a communication protocol. In this manner, computer-readable media generally may correspond to (1) tangible computer-readable storage media which is non-transitory or (2) a communication medium such as a signal or carrier wave. Data storage media may be any available media that can be accessed by one or more computers or one or more processors to retrieve instructions, code and/or data structures for implementation of the techniques described in this disclosure. A computer program product may include a computer-readable medium.

[0114] By way of example, and not limitation, such computer-readable storage media can comprise RAM, ROM, EEPROM, CD-ROM or other optical disk storage, magnetic disk storage, or other magnetic storage devices, flash memory, or any other medium that can be used to store desired program code in the form of instructions or data structures and that can be accessed by a computer. Also, any connection is properly termed a computer-readable medium. For example, if instructions are transmitted from a website, server, or other remote source using a coaxial cable, fiber optic cable, twisted pair, digital subscriber line (DSL), or wireless technologies such as infrared, radio, and microwave, then the coaxial cable, fiber optic cable, twisted pair, DSL, or wireless technologies such as infrared, radio, and microwave are included in the definition of medium. It should be understood, however, that computer-readable storage media and data storage media do not include connections, carrier waves, signals, or other transient media, but are instead directed to non-transient, tangible storage media. Disk and disc, as used herein, includes compact disc (CD), laser disc, optical disc, digital versatile disc (DVD), floppy disk and Blu-ray disc, where disks usually reproduce data magnetically, while discs reproduce data optically with lasers. Combinations of the above should also be included within the scope of computer-readable media.

[0115] Instructions may be executed by one or more processors, such as one or more digital signal processors (DSPs), general purpose microprocessors, application specific integrated circuits (ASICs), field programmable logic arrays (FPGAs), or other equivalent integrated or discrete logic circuitry. Accordingly, the term "processor," as used herein may refer to any of the foregoing structure or any other structure suitable for implementation of the techniques described herein. In addition, in some aspects, the functionality described herein may be provided within dedicated hardware and/or software modules configured for encoding and decoding, or incorporated in a combined codec. Also, the techniques could be fully implemented in one or more circuits or logic elements.

[0116] The techniques of this disclosure may be implemented in a wide variety of devices or apparatuses, including a wireless handset, an integrated circuit (IC) or a set of ICs (e.g., a chip set). Various components, modules, or units are described in this disclosure to emphasize functional aspects of devices configured to perform the disclosed techniques, but do not necessarily require realization by different hardware units. Rather, as described above, various units may be combined in a codec hardware unit or provided by a collection of interoperative hardware units, including one or more processors as described above, in conjunction with suitable software and/or firmware.

[0117] Various examples have been described. These and other examples are within the scope of the following claims.

[0118] It will be understood that the term "comprise" and any of its derivatives (eg comprises, comprising) as used in this specification is to be taken to be inclusive of features to which it refers, and is not meant to exclude the presence of any additional features unless otherwise stated or implied.

[0119] The reference to any prior art in this specification is not, and should not be taken as, an acknowledgement of any form of suggestion that such prior art forms part of the common general knowledge.

WHAT IS CLAIMED IS:

1. A method for encoding sample adaptive offset values in a video encoding process comprising:

determining sample adaptive offset information for a current partition;

performing a sample adaptive offset process for the current partition using the determined sample adaptive offset information;

comparing the sample offset information of the current partition with offset information of one or more neighbor partitions;

encoding a merge syntax element in the case that the sample adaptive offset information of one of the one or more neighbor partitions is the same as the sample adaptive offset information of the current partition, the merge syntax element indicating that a decoder is to copy the sample adaptive offset information for the current partition from the sample adaptive offset information of the one of the one or more neighbor partitions.

2. The method of claim 1, wherein the merge syntax element indicates which one of the one or more neighbor partitions from which to copy the sample adaptive offset information for the current partition.

3. The method of claim 1 or 2, wherein the sample adaptive offset information includes an offset type and offset values.

4. The method of claim 1, wherein the merge syntax element includes a plurality of syntax elements, the plurality of syntax elements including a merge flag to indicate the copying of the sample adaptive offset information for the current partition, and an index to indicate the specified neighbor partitions from which to copy the sample adaptive offset information for the current partition.

5. The method of claim 1, further including,

encoding a prediction syntax element in the case that the sample offset information of the one or more neighbor partitions is not the same as the sample offset information of the current partition.

6. The method of claim 5, wherein the prediction syntax element instructs the decoder to derive the sample adaptive offset information for the current partition from the sample adaptive offset information of one of the one or more neighbor partitions.

7. A method of decoding sample adaptive offset values in a video decoding process comprising:

obtaining from an encoded video bitstream, for a current partition, a merge syntax element indicating that sample adaptive offset information is to be copied from a specified neighbor partition;

copying sample adaptive offset information for the current partition from the specified neighbour partition in the case that the merge syntax element is obtained; and

performing a sample adaptive offset prediction process for the current partition using the copied sample adaptive offset information.

8. The method of claim 7, wherein copying the sample adaptive offset information includes copying an offset type and offset values from the specified neighbor partition.

9. The method of claim 7, wherein the merge syntax element includes data indicative of the specified neighbor partition and wherein the specified neighbor partition is one of a plurality of neighbor partitions.

10. The method of claim 7, further including:

obtaining from the encoded video bitstream, for another partition, a prediction instruction;

performing an offset prediction process to predict sample adaptive offset information for the another partition based on the prediction instruction; and

performing a sample adaptive offset process for the another partition using the predicted sample adaptive offset information.

11. The method of claim 7, wherein the merge syntax element includes a plurality of syntax elements, the plurality of syntax elements including a merge flag to indicate the copying of the sample adaptive offset information for the current partition, and an index

to indicate the specified neighbor partition from which to copy the sample adaptive offset information for the current partition.

12. The method of claim 7, wherein the merge syntax element includes a merge flag to indicate the copying of the sample adaptive offset information for the current partition.

13. The method of claim 9, wherein the sample adaptive offset information includes an offset type and offset values.

14. An apparatus for decoding sample adaptive offset information in a video decoding process including:

a memory configured to store data from an encoded video bitstream; and

a processor in communication with the memory and configured to:

obtain from an encoded video bitstream, for a current partition, a merge syntax element indicating that sample adaptive offset information is to be copied from a specified neighbor partition;

copy sample adaptive offset information for the current partition from the specified neighbor partition in the case that the merge syntax element is obtained; and

perform a sample adaptive offset process for the current partition using the copied sample adaptive offset information.

15. The apparatus of claim 14, wherein to copy the sample adaptive offset information, the processor is configured to copy an offset type and offset values from the specified neighbor partition.

16. The apparatus of claim 14, wherein the merge syntax element comprises data indicative of the specified neighbor partition and wherein the specified neighbor partition is one of a plurality of neighbor partitions.

17. The apparatus of claim 15, wherein the sample adaptive offset information includes an offset type and offset values.

18. The apparatus of claim 14, wherein the processor is configured to:

obtain from the encoded video bitstream, for another partition, a prediction instruction;

perform an offset prediction process to predict sample adaptive offset information for the another partition based on the prediction instruction; and

perform a sample adaptive offset process for the another partition using the predicted sample adaptive offset information.

19. The apparatus of claim 14, wherein the merge syntax element comprises a plurality of syntax elements, the plurality of syntax elements including a merge flag to indicate the copying of the sample adaptive offset information for the current partition, and an index to indicate the specified neighbor partition from which to copy the sample adaptive offset information for the current partition.

20. A computer program product comprising a computer-readable storage medium having stored thereon instructions that, when executed, cause a processor to perform a method according to any one of claims 1 to 13.

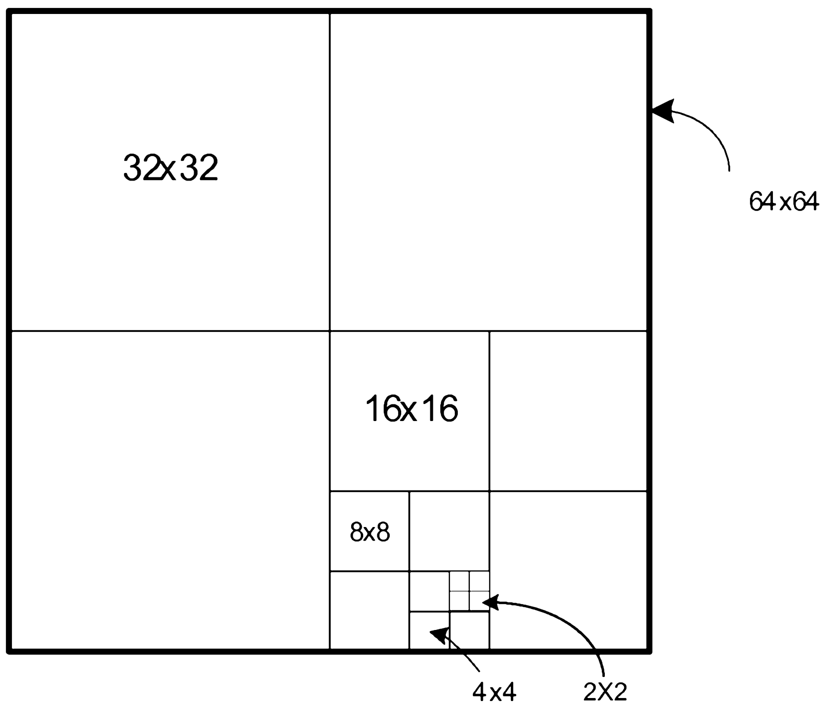
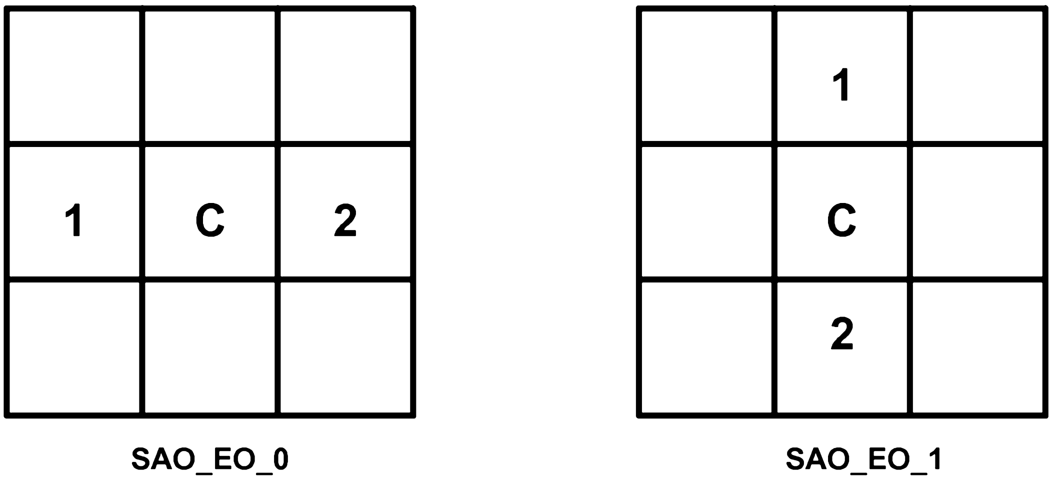


FIG. 1



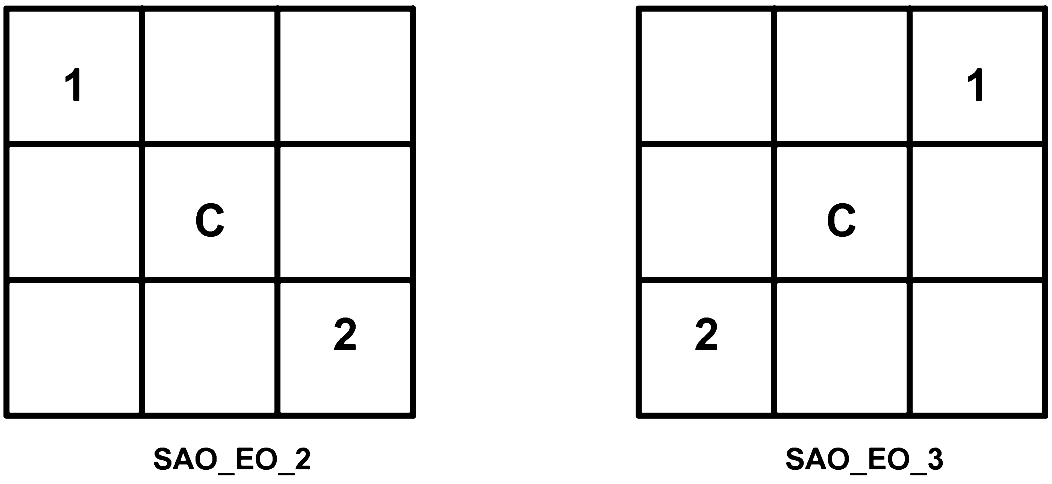


FIG. 2

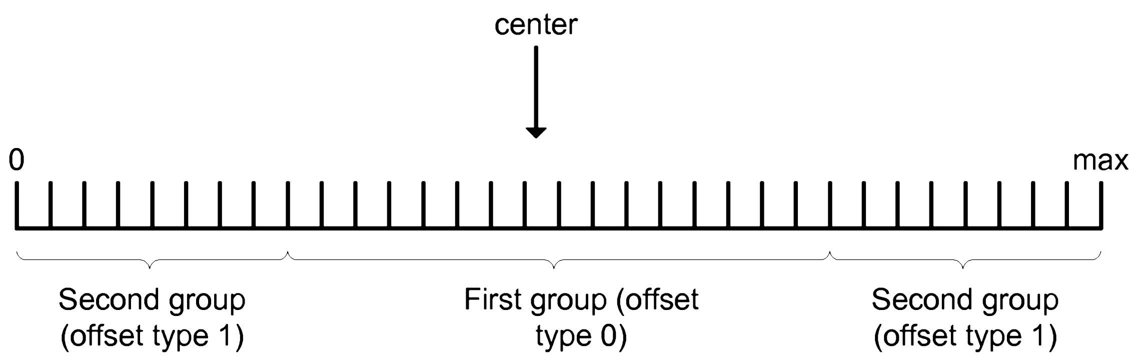

center
0
max
Second group (offset type 1)
First group (offset type 0)
Second group (offset type 1)

FIG. 3

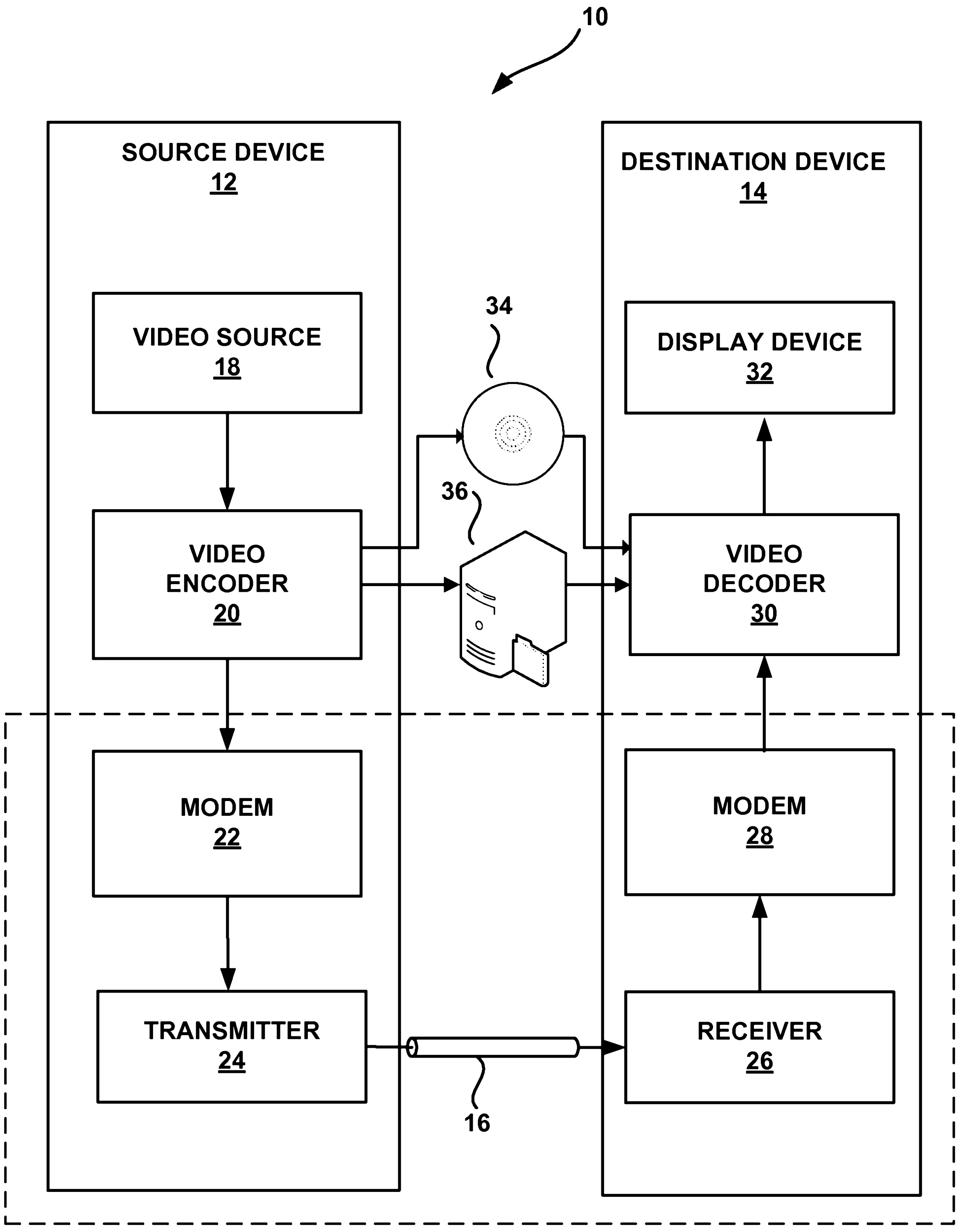

10
SOURCE DEVICE 12
VIDEO SOURCE 18
VIDEO ENCODER 20
MODEM 22
TRANSMITTER 24
34
36
16
DESTINATION DEVICE 14
DISPLAY DEVICE 32
VIDEO DECODER 30
MODEM 28
RECEIVER 26

FIG. 4

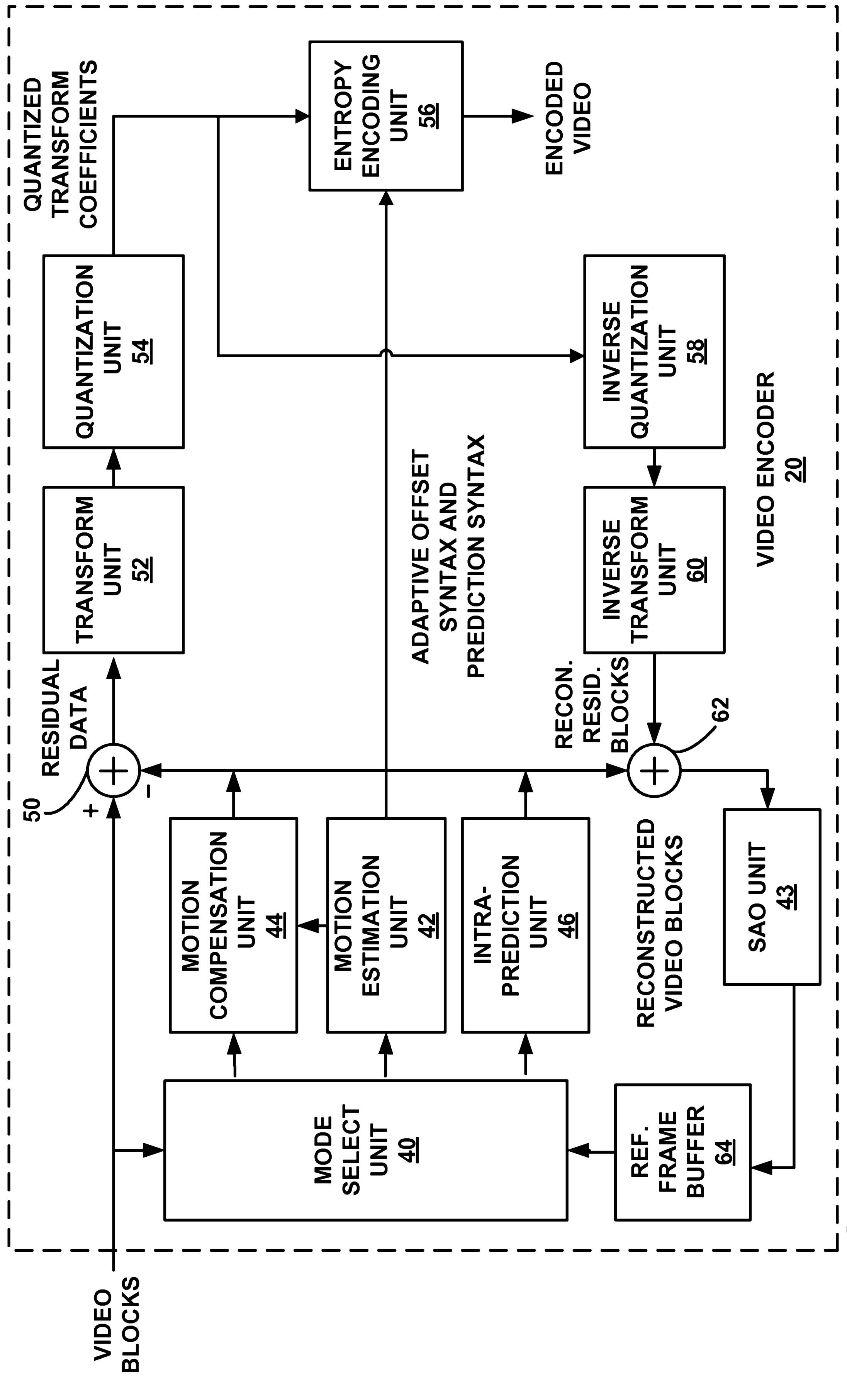

VIDEO BLOCKS
MODE SELECT UNIT 40
MOTION COMPENSATION UNIT 44
MOTION ESTIMATION UNIT 42
INTRA-PREDICTION UNIT 46
50
+
−
RESIDUAL DATA
TRANSFORM UNIT 52
QUANTIZATION UNIT 54
QUANTIZED TRANSFORM COEFFICIENTS
ENTROPY ENCODING UNIT 56
ENCODED VIDEO
ADAPTIVE OFFSET SYNTAX AND PREDICTION SYNTAX
INVERSE QUANTIZATION UNIT 58
INVERSE TRANSFORM UNIT 60
RECON. RESID. BLOCKS
62
RECONSTRUCTED VIDEO BLOCKS
SAO UNIT 43
REF. FRAME BUFFER 64
VIDEO ENCODER 20

FIG. 5

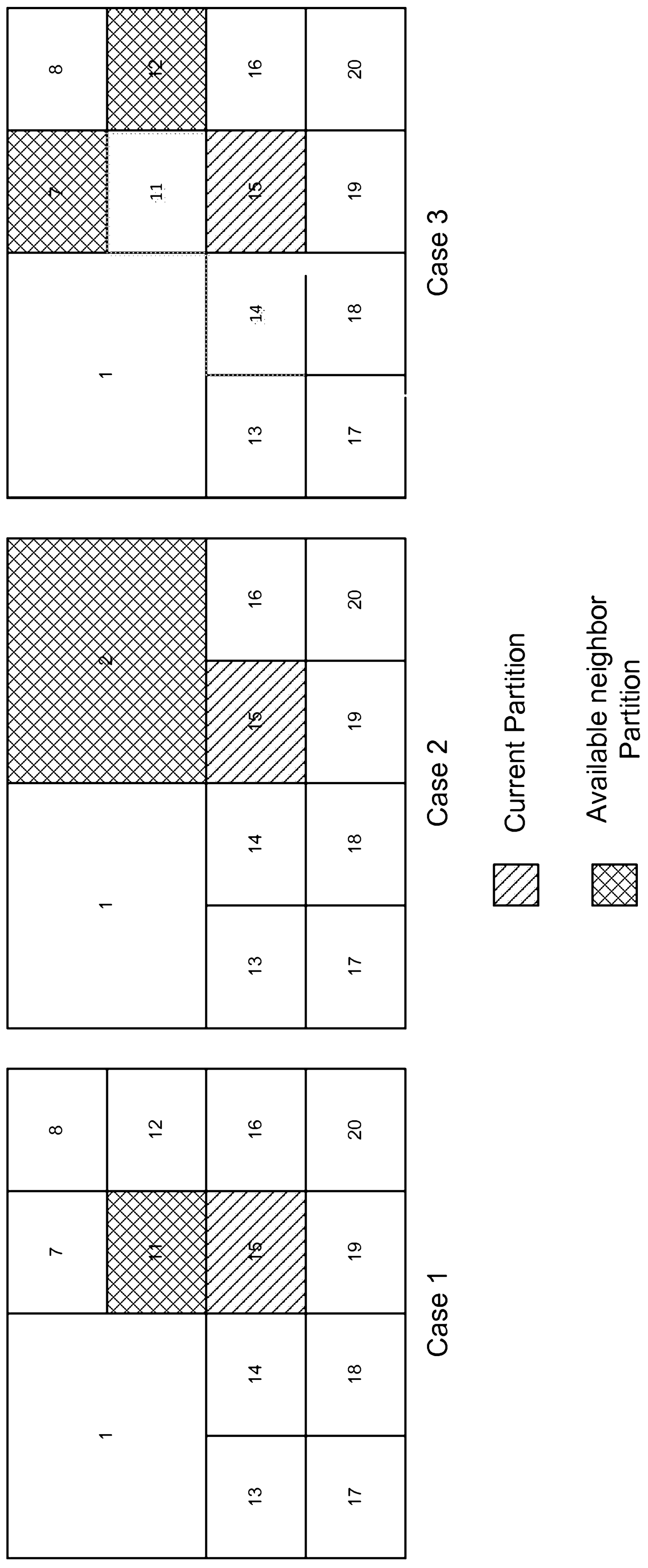

1
7
8
11
12
13
14
15
16
17
18
19
20
Case 1
1
2
13
14
15
16
17
18
19
20
Case 2
1
7
8
11
12
13
14
15
16
17
18
19
20
Case 3
Current Partition
Available neighbor Partition

FIG. 6

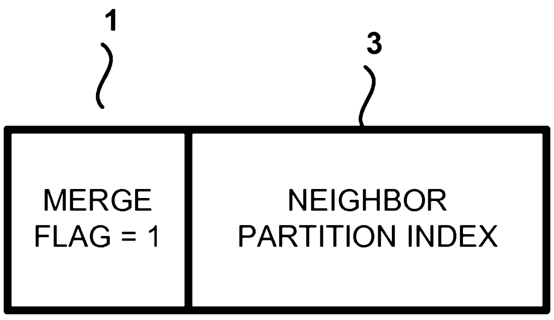

1
3
MERGE FLAG = 1
NEIGHBOR PARTITION INDEX

FIG. 7A

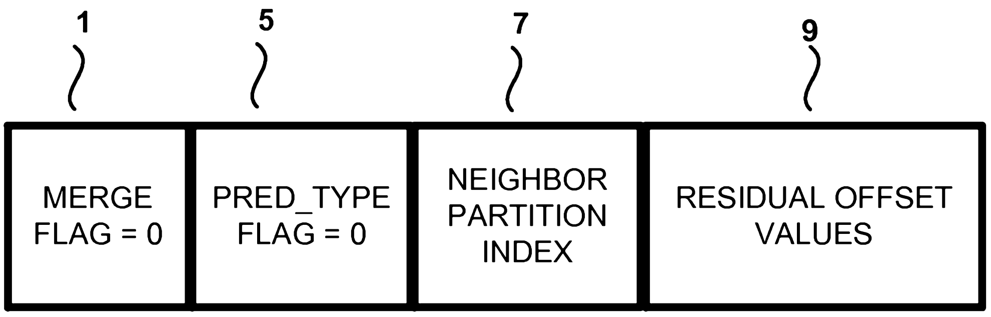

1
5
7
9
MERGE FLAG = 0
PRED_TYPE FLAG = 0
NEIGHBOR PARTITION INDEX
RESIDUAL OFFSET VALUES

FIG. 7B

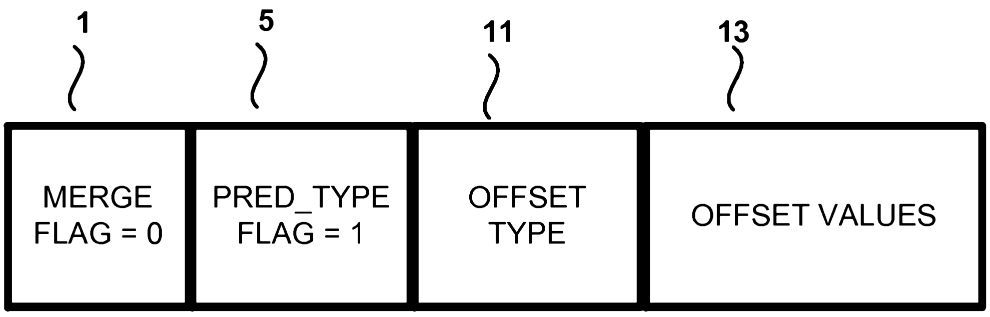

1
5
11
13
MERGE FLAG = 0
PRED_TYPE FLAG = 1
OFFSET TYPE
OFFSET VALUES

FIG. 7C

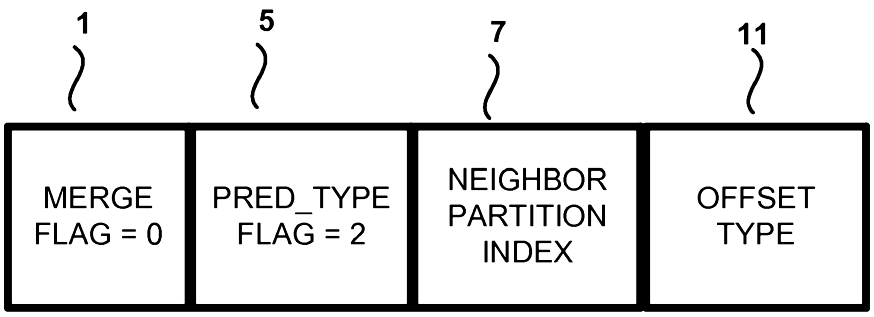

1
5
7
11
MERGE FLAG = 0
PRED_TYPE FLAG = 2
NEIGHBOR PARTITION INDEX
OFFSET TYPE

FIG. 7D

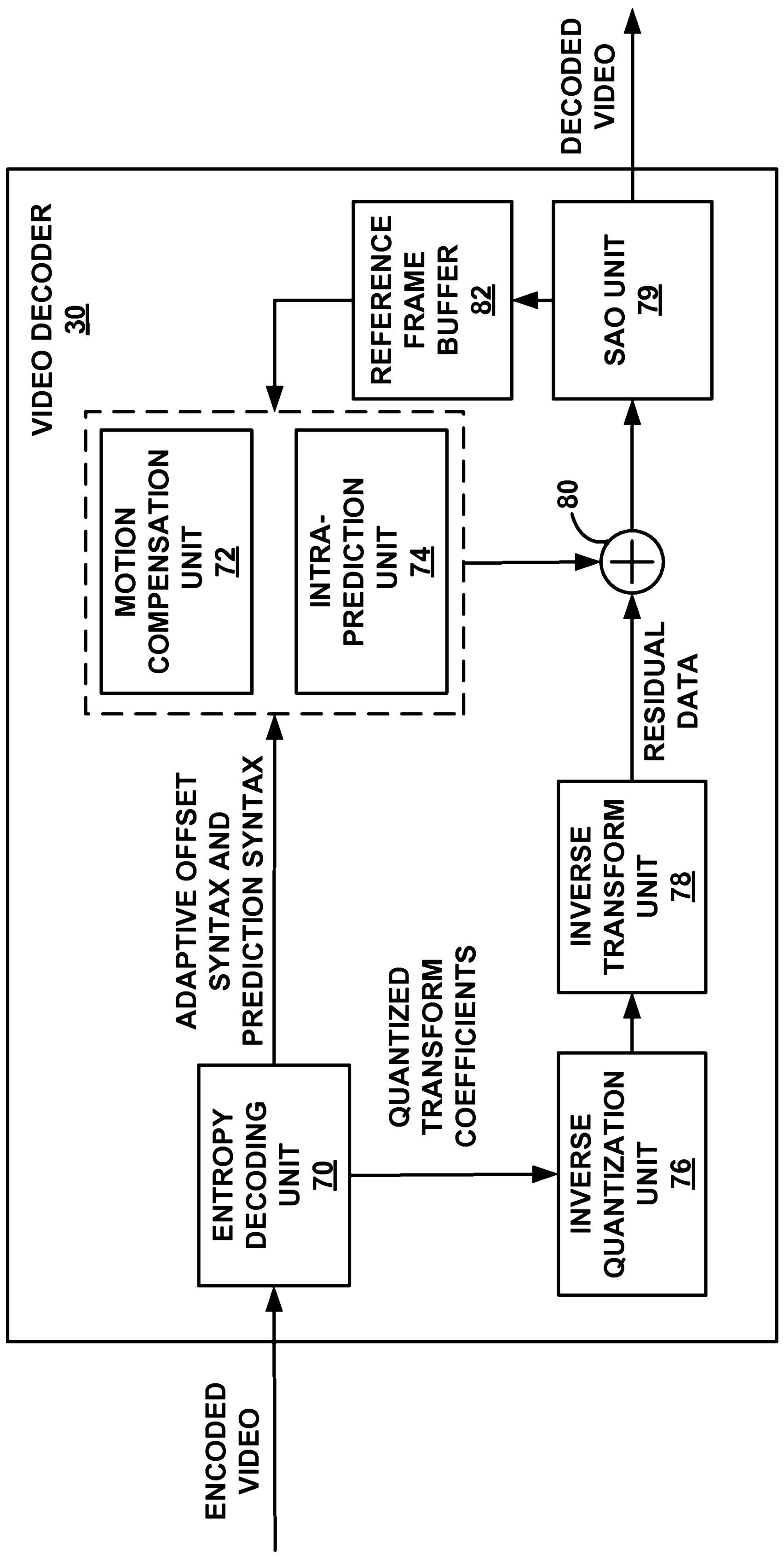

VIDEO DECODER 30
ENCODED VIDEO
ENTROPY DECODING UNIT 70
ADAPTIVE OFFSET SYNTAX AND PREDICTION SYNTAX
QUANTIZED TRANSFORM COEFFICIENTS
MOTION COMPENSATION UNIT 72
INTRA-PREDICTION UNIT 74
INVERSE QUANTIZATION UNIT 76
INVERSE TRANSFORM UNIT 78
RESIDUAL DATA
80
SAO UNIT 79
REFERENCE FRAME BUFFER 82
DECODED VIDEO

FIG. 8

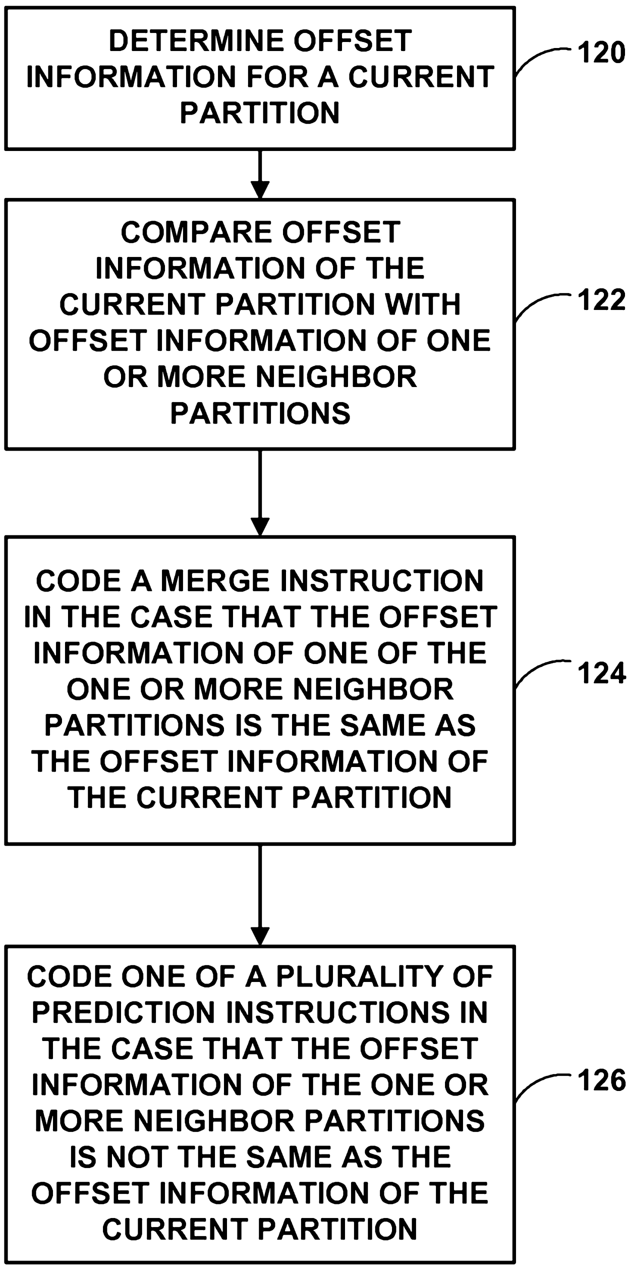

DETERMINE OFFSET INFORMATION FOR A CURRENT PARTITION
120
COMPARE OFFSET INFORMATION OF THE CURRENT PARTITION WITH OFFSET INFORMATION OF ONE OR MORE NEIGHBOR PARTITIONS
122
CODE A MERGE INSTRUCTION IN THE CASE THAT THE OFFSET INFORMATION OF ONE OF THE ONE OR MORE NEIGHBOR PARTITIONS IS THE SAME AS THE OFFSET INFORMATION OF THE CURRENT PARTITION
124
CODE ONE OF A PLURALITY OF PREDICTION INSTRUCTIONS IN THE CASE THAT THE OFFSET INFORMATION OF THE ONE OR MORE NEIGHBOR PARTITIONS IS NOT THE SAME AS THE OFFSET INFORMATION OF THE CURRENT PARTITION
126

FIG. 9

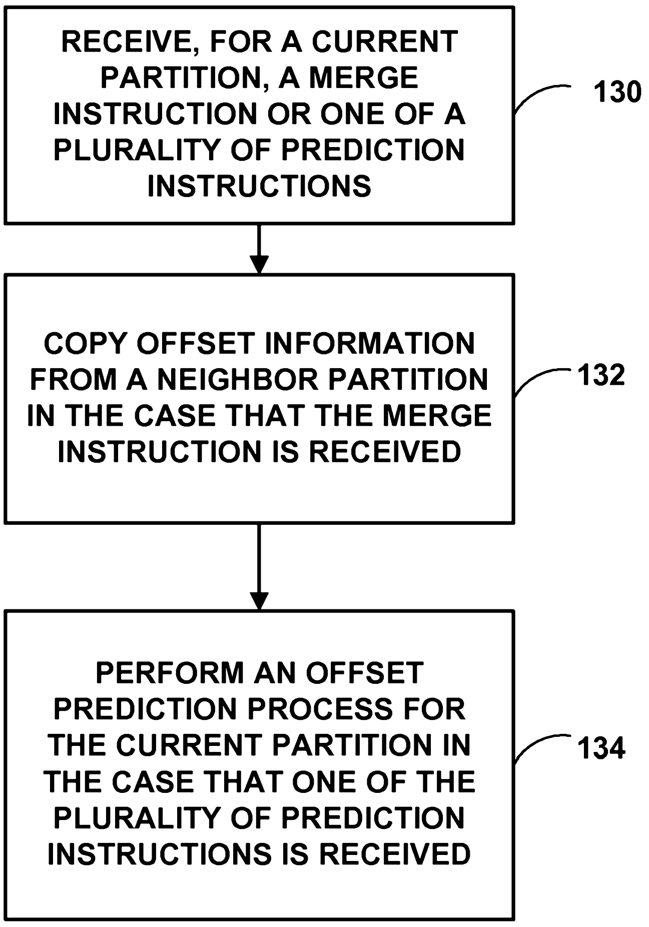

RECEIVE, FOR A CURRENT PARTITION, A MERGE INSTRUCTION OR ONE OF A PLURALITY OF PREDICTION INSTRUCTIONS
130
COPY OFFSET INFORMATION FROM A NEIGHBOR PARTITION IN THE CASE THAT THE MERGE INSTRUCTION IS RECEIVED
132
PERFORM AN OFFSET PREDICTION PROCESS FOR THE CURRENT PARTITION IN THE CASE THAT ONE OF THE PLURALITY OF PREDICTION INSTRUCTIONS IS RECEIVED
134

FIG. 10