



Office de la Propriété

Intellectuelle
du Canada

Un organisme
d'Industrie Canada

Canadian
Intellectual Property
Office

An agency of
Industry Canada

CA 2918762 A1 2015/03/05

(21) **2 918 762**

(12) **DEMANDE DE BREVET CANADIEN**
CANADIAN PATENT APPLICATION

(13) **A1**

(86) Date de dépôt PCT/PCT Filing Date: 2014/08/27
(87) Date publication PCT/PCT Publication Date: 2015/03/05
(85) Entrée phase nationale/National Entry: 2016/01/19
(86) N° demande PCT/PCT Application No.: US 2014/052955
(87) N° publication PCT/PCT Publication No.: 2015/031499
(30) Priorités/Priorities: 2013/08/27 (US61/870,654);
2013/08/28 (US61/871,252); 2014/08/26 (US14/469,477)

(51) CI.Int./Int.Cl. *H04N 19/70* (2014.01),
H04N 19/593 (2014.01)

(71) Demandeur/Applicant:
QUALCOMM INCORPORATED, US

(72) Inventeurs/Inventors:
GUO, LIWEI, US;
PANG, CHAO, US;
JOSHI, RAJAN LAXMAN, US;
SOLE ROJALS, JOEL, US;
KARCZEWCZ, MARTA, US

(74) Agent: SMART & BIGGAR

(54) Titre : PREDICTION RESIDUELLE POUR UNE COPIE INTRA-BLOC

(54) Title: RESIDUAL PREDICTION FOR INTRA BLOCK COPYING

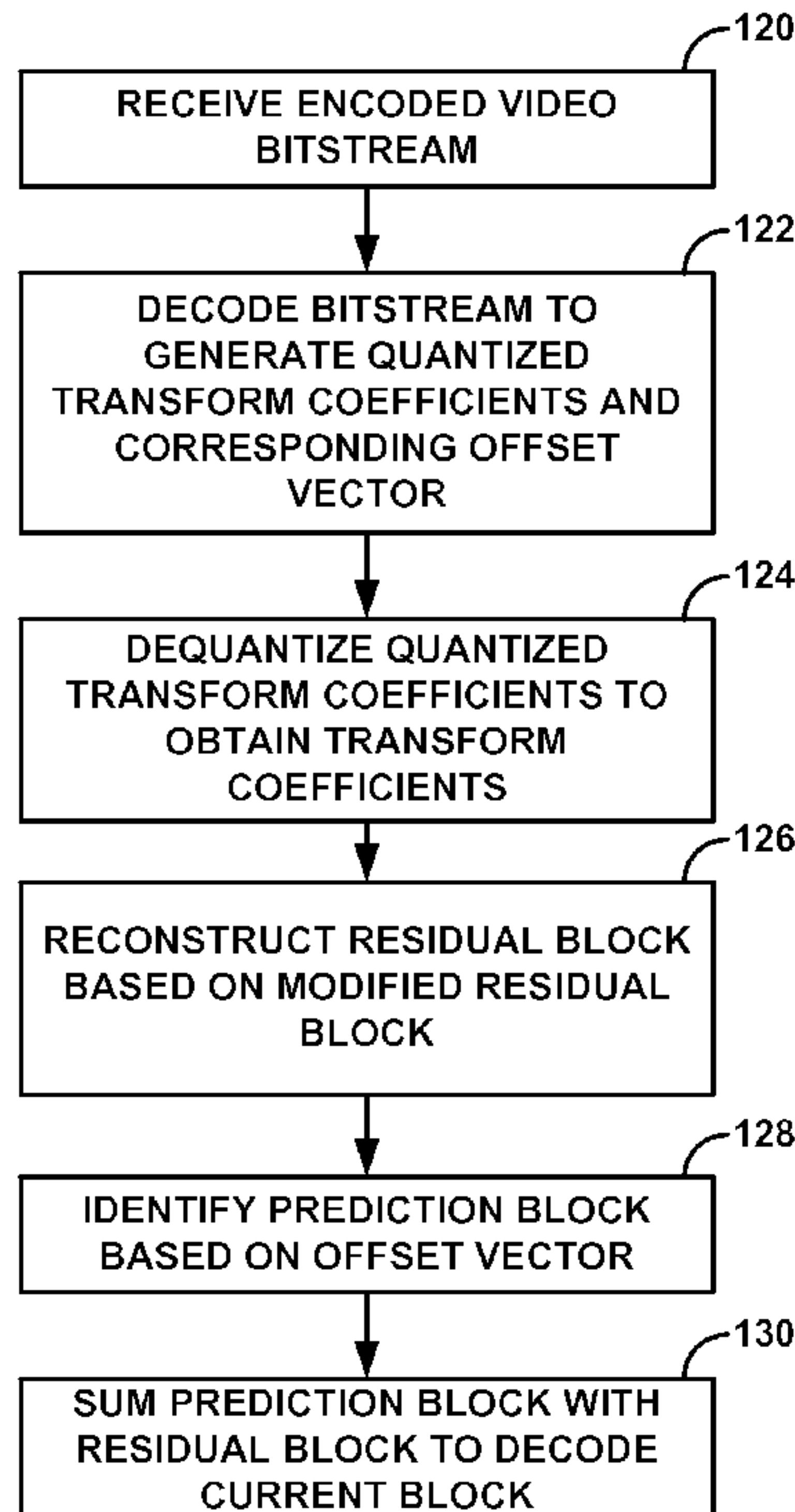


FIG. 7

(57) Abrégé/Abstract:

In an example, a method of decoding video data includes generating a residual block of a picture based on a predicted residual block including reconstructing one or more residual values of the residual block based on one or more predicted residual values of the residual block. The method also includes generating a current block of the picture based on a combination of the residual block and a prediction block of the picture.

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

(19) World Intellectual Property Organization
International Bureau



(10) International Publication Number
WO 2015/031499 A1

(43) International Publication Date
5 March 2015 (05.03.2015)

(51) International Patent Classification:
H04N 19/70 (2014.01) H04N 19/593 (2014.01)

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

(21) International Application Number:
PCT/US2014/052955

(72) Inventors: **GUO, Liwei**; 5230 Fiore Ter. 303, San Diego, California 92122 (US). **PANG, Chao**; 5775 Morehouse Drive, San Diego, California 92121-1714 (US). **JOSHI, Rajan Laxman**; 5775 Morehouse Drive, San Diego, California 92121-1714 (US). **SOLE ROJALS, Joel**; 5775 Morehouse Drive, San Diego, California 92121-1714 (US). **KARCZEWCZ, Marta**; 5775 Morehouse Drive, San Diego, California 92121-1714 (US).

(22) International Filing Date:
27 August 2014 (27.08.2014)

(74) Agent: **PRIEM, David F.**; Shumaker & Sieffert, P.A., 1625 Radio Drive, Suite 300, Woodbury, Minnesota 55125 (US).

(25) Filing Language:
English

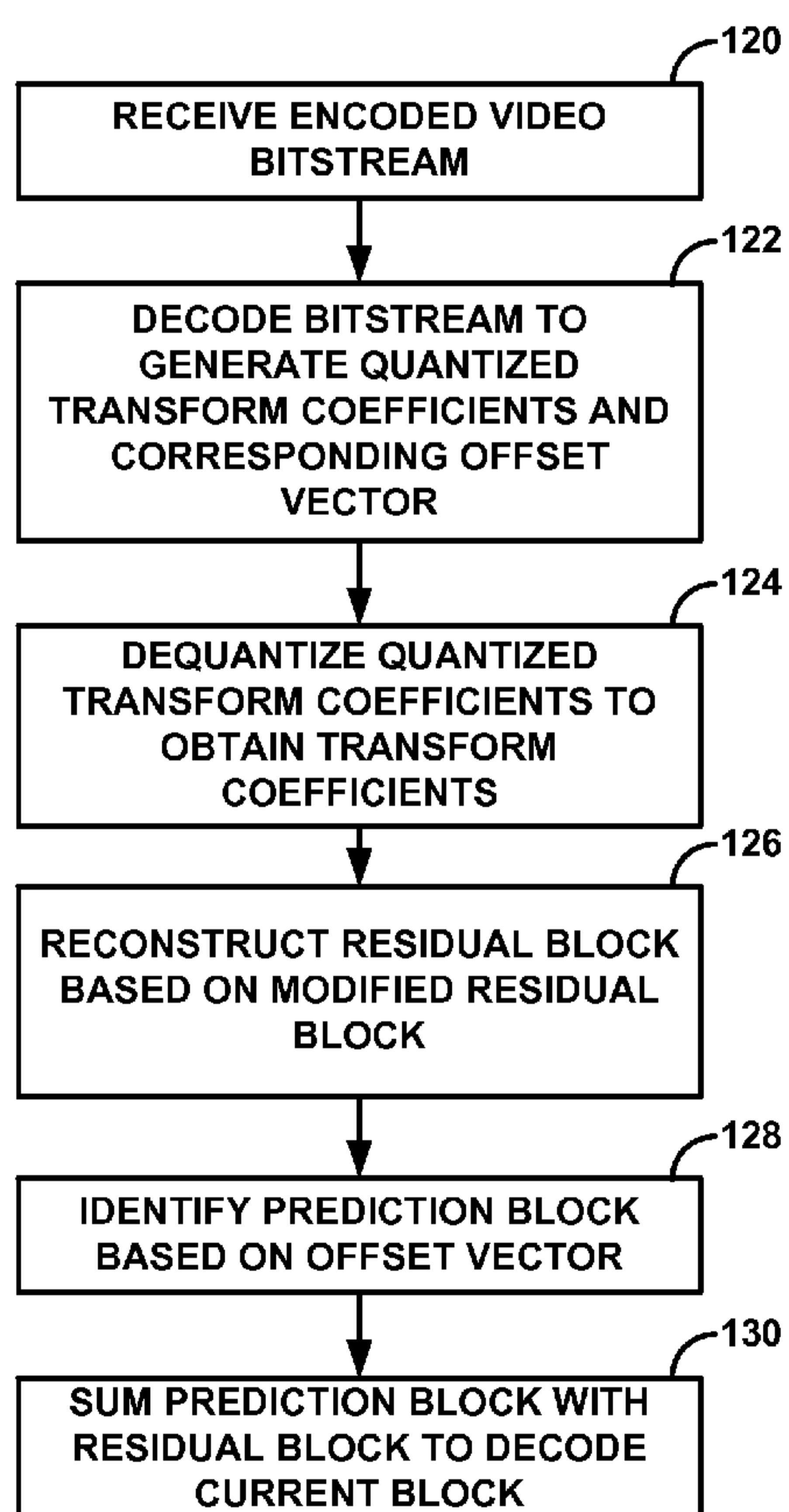
(26) Publication Language:
English

(30) Priority Data:
61/870,654 27 August 2013 (27.08.2013)
61/871,252 28 August 2013 (28.08.2013)
14/469,477 26 August 2014 (26.08.2014)

US
US
US

[Continued on next page]

(54) Title: RESIDUAL PREDICTION FOR INTRA BLOCK COPYING



(57) **Abstract:** In an example, a method of decoding video data includes generating a residual block of a picture based on a predicted residual block including reconstructing one or more residual values of the residual block based on one or more predicted residual values of the residual block. The method also includes generating a current block of the picture based on a combination of the residual block and a prediction block of the picture.

WO 2015/031499 A1

FIG. 7

WO 2015/031499 A1



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

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

Published:

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

RESIDUAL PREDICTION FOR INTRA BLOCK COPYING

[0001] This application claims the benefit of U.S. Provisional Patent Application No. 61/870,654, filed August 27, 2013, and U.S. Provisional Patent Application No. 61/871,252, filed August 28, 2013, the entire contents of which are incorporated herein by reference.

TECHNICAL FIELD

[0002] This disclosure relates to video coding, and more particularly to techniques for predicting video data.

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 picture or slice may be partitioned into blocks. Each block can be further partitioned. Blocks in an intra-coded (I) picture or slice are encoded using spatial prediction with respect to reference samples in neighboring blocks in the same picture or slice. Blocks in an inter-coded (P or B) picture or slice may use spatial prediction with respect to reference samples in neighboring blocks in the same picture or slice or temporal prediction with respect to reference samples in other reference pictures. Spatial or temporal prediction results in a predictive block (which may also be referred to as a “prediction block”) for a block to be coded. 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.

SUMMARY

[0006] Techniques of this disclosure generally relate to predicting data in video coding. For example, the techniques described in this disclosure may provide support for applying both intra-block copying (intra-BC) and residue differential pulse code modulation (RDPCM) in video coding. In an example, the techniques of this disclosure may include predicting residual data of an intra-BC predicted block using RDPCM.

[0007] In an example, a method of decoding video data includes generating a residual block of a picture based on a predicted residual block including reconstructing one or more residual values of the residual block based on one or more predicted residual values of the residual block, and generating a current block of the picture based on a combination of the residual block and a prediction block of the picture.

[0008] In another example, a method of encoding video data includes generating a residual block for a current block of a picture based on a difference between the current block and a prediction block of the picture, generating a predicted residual block based on the residual block by predicting one or more residual values of the residual block based on one or more other residual values of the residual block, and encoding data that represents the predicted residual block in a bitstream.

[0009] In another example, device for of decoding video data includes a memory storing the video data, and a video decoder configured to generate a residual block of a picture based on a predicted residual block including reconstructing one or more residual values of the residual block based on one or more predicted residual values of the residual block, and generate a current block of the picture based on a combination of the residual block and a prediction block of the picture.

[0010] In another example, a device for encoding video data includes a memory storing the video data, and a video encoder configured to generate a residual block for a current block of a picture based on a difference between the current block and a prediction

block of the picture, generate a predicted residual block based on the residual block by predicting one or more residual values of the residual block based on one or more other residual values of the residual block, and encode data that represents the predicted residual block in a bitstream.

[0011] In another example, an device for decoding video data includes means for generating a residual block of a picture based on a predicted residual block including reconstructing one or more residual values of the residual block based on one or more predicted residual values of the residual block, and means for generating a current block of the picture based on a combination of the residual block and a prediction block of the picture.

[0012] In another example, a device for encoding video data includes means for generating a residual block for a current block of a picture based on a difference between the current block and a prediction block of the picture, means for generating a predicted residual block based on the residual block by predicting one or more residual values of the residual block based on one or more other residual values of the residual block, and means for encoding data that represents the predicted residual block in a bitstream.

[0013] In another example, a non-transitory computer-readable medium has instructions stored thereon that, when executed, cause one or more processors to generate a residual block of a picture based on a predicted residual block including reconstructing one or more residual values of the residual block based on one or more predicted residual values of the residual block, and generate a current block of the picture based on a combination of the residual block and a prediction block of the picture.

[0014] In another example, a non-transitory computer-readable medium has instructions stored thereon that, when executed, cause one or more processors to generate a residual block for a current block of a picture based on a difference between the current block and a prediction block of the picture, generate a predicted residual block based on the residual block by predicting one or more residual values of the residual block based on one or more other residual values of the residual block, and encode data that represents the predicted residual block in a bitstream.

[0015] The details of one or more aspects of the disclosure are set forth in the accompanying drawings and the description below. Other features, objects, and advantages of the techniques described in this disclosure will be apparent from the description and drawings, and from the claims.

BRIEF DESCRIPTION OF DRAWINGS

[0016] FIG. 1 is a block diagram illustrating an example video encoding and decoding system that may implement the techniques of this disclosure.

[0017] FIG. 2 is a block diagram illustrating an example video encoder that may implement the techniques of this disclosure.

[0018] FIG. 3 is a block diagram illustrating an example video decoder that may implement the techniques of this disclosure.

[0019] FIG. 4 is a diagram illustrating an example of an intra-block copying (intra-BC) process.

[0020] FIG. 5A shows a residual DPCM direction for near-vertical modes.

[0021] FIG. 5B shows a residual DPCM direction for near-horizontal modes.

[0022] FIG. 6 is a flowchart illustrating an example technique of encoding video data in accordance with this disclosure.

[0023] FIG. 7 is a flowchart illustrating an example technique of decoding video data in accordance with this disclosure.

DETAILED DESCRIPTION

[0024] Aspects of this disclosure generally relate to video coding and compression. In some examples, the techniques may be related to a High Efficiency Video Coding (HEVC) Range Extension. HEVC is a video coding standard that was recently developed by the Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T WP3/16 and ISO/IEC JTC 1/SC 29/WG 11. In general, the HEVC Range Extension may support video formats that are not specifically supported by the base HEVC specification.

[0025] The techniques may also relate to an HEVC screen contents coding extension to HEVC. More information for the screen contents coding extension of HEVC can be found in the document of the Visual Coding group of ITU-T Q6/16 and ISO/IEC/JTC1/SC29/WG11, entitled “Joint Call for Proposals for Coding of Screen Content,” presented in the 17 January 2014 meeting at San Jose, California and available at: <http://www.itu.int/en/ITU-T/studygroups/com16/video/Documents/CfP-HEVC-coding-screen-content.pdf>.

[0026] The Range Extension of HEVC, the screen contents coding extension, or other video coding tools may support intra-block copying (intra-BC) and residue differential pulse code modulation (RDPCM). With respect to intra-BC, for many applications, such as remote desktop, remote gaming, wireless displays, automotive infotainment, and cloud computing to provide a few examples, the video contents in these applications are usually combinations of natural content, text, artificial graphics and the like. In test and artificial graphics regions, repeated patterns (such as characters, icons, and symbols to provide a few examples) often exist. Intra-BC may be characterized as a dedicated process to removal of this kind of redundancy, thereby potentially improving the intra-frame coding efficiency.

[0027] With respect to RDPCM, a video coder (such as a video encoder or video decoder) may apply RDPCM to intra-prediction residual (e.g., residual predicted using one of the 35 directional HEVC intra-modes) or to motion compensated prediction residual (e.g., residual predicted using temporal motion prediction) to reduce the amount of data of a residual block that needs to be output for the residual block. In RDPCM, the video encoder predicts the residual values of a row or column of a residual block based on the residual values of the previous row or column. In general, RDPCM may include three different modes: RDPCM-off, RDPCM-horizontal and RDPCM-vertical. For RDPCM-off, the video coder may not apply RDPCM. For RDPCM horizontal, the video coder may use the residual samples in the left column to predict the residual samples in the current column. For RDPCM vertical, the video coder may use residual samples in the upper row to predict the residual samples in the current row.

[0028] As an example, to perform vertical RDPCM, the video encoder subtracts the residual values of row 1 from the reconstructed residual values of row 0 (e.g., subtract the first residual value of row 1 from the first residual value of row 0, subtract the second residual value of row 1 from the second residual value of row 0, and so forth). For row 0, the video encoder signals the residual values, but for row 1, the video encoder signals the difference values, and for row 2, the video encoder signals the difference values with the reconstructed values of row 1, and so forth. The video encoder may perform a similar technique for horizontal based RDPCM. Subtraction, as described in this disclosure, refers to determining values equal to the result of subtracting, and can be performed by subtracting or adding a negative of value.

[0029] In some examples, encoding the difference between residual values of successive rows or columns may result in fewer bits than encoding the actual residual

values. In this manner, RDPCM may result in a reduction in the amount of data the video encoder needs to signal, thereby promoting bandwidth efficiency.

[0030] In the example above, the video decoder decodes the received values. For instance, the video decoder decodes the residual values for row 0, and decodes the difference values for row 1. For example, the video decoder may dequantize the received values for row 0 and row 1. The video decoder adds the residual values for row 0 with the difference values to determine the residual values for row 1. The video decoder also decodes the difference values for row 2 (e.g., the difference between the residual values of row 1 and the residual values of row 2). The video decoder adds the determined residual values for row 1 with the difference values of row 2 to determine the residual values of row 2, and so forth. The video decoder may implement similar steps for horizontal based RDPCM.

[0031] In some examples, the video encoder may utilize RDPCM only in conjunction with certain other coding tools. For example, the video encoder may utilize lossless (also referred to as bypass) or transform-skip coding to generate the residual block or a quantized version of the residual block, respectively. In some examples, the video encoder may utilize RDPCM only if the residual block is transform-bypass or transform-skip encoded. If a transform is applied to the residual block, then RDPCM may not be available to the video encoder.

[0032] At the JCTVC meeting in Vienna (July 2013), both intra-BC and RDPCM were adopted in the HEVC Range Extension standard noted above. However, there were no techniques for reducing an amount of residual data being signaled when using an intra-BC mode to generate prediction residual. That is, there were no techniques for applying RDPCM to residual data generated using an intra-BC mode.

[0033] The techniques described in this disclosure provide support for applying both RDPCM and intra-BC in video coding. For example, the techniques of this disclosure may include using an intra-BC prediction mode to generate residual data, and predicting the residual data of the intra-BC predicted block using RDPCM, as discussed in further detail below. The techniques of this disclosure may further improve efficiency when applying intra-BC techniques. For example, by applying RDPCM to a block that has been predicted using intra-BC techniques, residual data may be further decreased, thereby reducing an amount of data that is included in an encoded bitstream and increasing overall coding efficiency. That is, the prediction block described above (in which RDPCM is applied to intra-BC prediction residual) may require fewer bits to

represent in a bitstream than a residual block predicted using an intra-BC mode without applying RDPCM.

[0034] FIG. 1 is a block diagram illustrating an example video encoding and decoding system 10 that may utilize techniques for filtering video data. As shown in FIG. 1, system 10 includes a source device 12 that provides encoded video data to be decoded at a later time by a destination device 14. In particular, source device 12 provides the video data to destination device 14 via a computer-readable medium 16. Source device 12 and destination device 14 may comprise any of a wide range of devices, including desktop computers, notebook (i.e., 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, source device 12 and destination device 14 may be equipped for wireless communication.

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

[0036] In some examples, encoded data may be output from output interface 22 to a storage device. Similarly, encoded data may be accessed from the storage device by input interface. The storage device may include any of a variety of distributed or locally accessed data storage media such as a hard drive, Blu-ray discs, DVDs, CD-ROMs, flash memory, volatile or non-volatile memory, or any other suitable digital storage media for storing encoded video data. In a further example, the storage device may

correspond to a file server or another intermediate storage device that may store the encoded video generated by source device 12.

[0037] Destination device 14 may access stored video data from the storage device via streaming or download. The file server may be any type of server capable of storing encoded video data and transmitting that encoded video data to the destination device 14. Example file servers include a web server (e.g., for a website), an FTP server, network attached storage (NAS) devices, or a local disk drive. Destination device 14 may access the encoded video data through any standard data connection, including an Internet connection. This may include a wireless channel (e.g., a Wi-Fi connection), a wired connection (e.g., DSL, cable modem, etc.), or a combination of both that is suitable for accessing encoded video data stored on a file server. The transmission of encoded video data from the storage device may be a streaming transmission, a download transmission, or a combination thereof.

[0038] The techniques of this disclosure are not necessarily 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, Internet streaming video transmissions, such as dynamic adaptive streaming over HTTP (DASH), digital video that is encoded onto a data storage medium, decoding of digital video stored on a data storage medium, or other applications. In some examples, system 10 may be configured to support one-way or two-way video transmission to support applications such as video streaming, video playback, video broadcasting, and/or video telephony.

[0039] In the example of FIG. 1, source device 12 includes video source 18, video encoder 20, and output interface 22. Destination device 14 includes input interface 28, video decoder 30, and display device 32. In accordance with this disclosure, video encoder 20 of source device 12 may be configured to apply the techniques for performing transformation in video coding. In other examples, a source device and a destination device may include other components or arrangements. For example, source device 12 may receive video data from an external video source 18, such as an external camera. Likewise, destination device 14 may interface with an external display device, rather than including an integrated display device.

[0040] The illustrated system 10 of FIG. 1 is merely one example. Techniques for performing filtering in video coding may be performed by any digital video encoding and/or decoding device. Although generally the techniques of this disclosure are

performed by a video encoding device, the techniques may also be performed by a video codec. Moreover, the techniques of this disclosure may also be performed by a video preprocessor. Source device 12 and destination device 14 are merely examples of such coding devices in which source device 12 generates coded video data for transmission to destination device 14. In some examples, devices 12, 14 may operate in a substantially symmetrical manner such that each of devices 12, 14 include video encoding and decoding components. Hence, system 10 may support one-way or two-way video transmission between video devices 12, 14, e.g., for video streaming, video playback, video broadcasting, or video telephony.

[0041] Video source 18 of source device 12 may include a video capture device, such as a video camera, a video archive containing previously captured video, and/or a video feed interface to receive video from a video content provider. As a further alternative, video source 18 may generate computer graphics-based data as the source video, or a combination of live video, archived video, and computer-generated video. In some cases, if video source 18 is a video camera, source device 12 and destination device 14 may form so-called camera phones or video phones. As mentioned above, 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. In each case, the captured, pre-captured, or computer-generated video may be encoded by video encoder 20. The encoded video information may then be output by output interface 22 onto a computer-readable medium 16.

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

[0043] Input interface 28 of destination device 14 receives information from computer-readable medium 16. The information of computer-readable medium 16 may include syntax information defined by video encoder 20, which is also used by video decoder

30, that includes syntax elements that describe characteristics and/or processing of blocks and other coded units, e.g., GOPs. Display device 32 displays the decoded video data to a user, and may comprise any of a variety of display devices such as a cathode ray tube (CRT), a liquid crystal display (LCD), a plasma display, an organic light emitting diode (OLED) display, or another type of display device.

[0044] Video encoder 20 and video decoder 30 each may be implemented as any of a variety of suitable encoder or decoder circuitry, as applicable, such as one or more microprocessors, digital signal processors (DSPs), application specific integrated circuits (ASICs), field programmable gate arrays (FPGAs), discrete logic circuitry, software, hardware, firmware or any combinations thereof. When the techniques are implemented partially in software, a device may store instructions for the software in a suitable, non-transitory computer-readable medium and execute the instructions in hardware using one or more processors to perform the techniques of this disclosure. Each of video encoder 20 and video decoder 30 may be included in one or more encoders or decoders, either of which may be integrated as part of a combined video encoder/decoder (codec). A device including video encoder 20 and/or video decoder 30 may comprise an integrated circuit, a microprocessor, and/or a wireless communication device, such as a cellular telephone.

[0045] Although not shown in FIG. 1, in some aspects, video encoder 20 and video decoder 30 may each be integrated with an audio encoder and decoder, and may include appropriate MUX-DEMUX units, or other hardware and software, to handle encoding of both audio and video in a common data stream or separate data streams. If applicable, MUX-DEMUX units may conform to the ITU H.223 multiplexer protocol, or other protocols such as the user datagram protocol (UDP).

[0046] This disclosure may generally refer to video encoder 20 “signaling” certain information to another device, such as video decoder 30. It should be understood, however, that video encoder 20 may signal information by associating certain syntax elements with various encoded portions of video data. That is, video encoder 20 may “signal” data by storing certain syntax elements to headers of various encoded portions of video data. In some cases, such syntax elements may be encoded and stored (e.g., stored to storage device 24) prior to being received and decoded by video decoder 30. Thus, the term “signaling” may generally refer to the communication of syntax or other data for decoding compressed video data, whether such communication occurs in real- or near-real-time or over a span of time, such as might occur when storing syntax

elements to a medium at the time of encoding, which then may be retrieved by a decoding device at any time after being stored to this medium.

[0047] Video encoder 20 and video decoder 30 may operate according to a video compression standard, 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 ITU-T H.264/MPEG-4 (AVC) standard was formulated by the ITU-T Video Coding Experts Group (VCEG) together with the ISO/IEC Moving Picture Experts Group (MPEG) as the product of a collective partnership known as the Joint Video Team (JVT). In some aspects, the techniques described in this disclosure may be applied to devices that generally conform to the H.264 standard. The H.264 standard is described in ITU-T Recommendation H.264, Advanced Video Coding for generic audiovisual services, by the ITU-T Study Group, and dated March, 2005, which may be referred to herein as the H.264 standard or H.264 specification, or the H.264/AVC standard or specification. Other examples of video compression standards include MPEG-2 and ITU-T H.263.

[0048] While the techniques of this disclosure are not limited to any particular coding standard, the techniques may be relevant to the HEVC standard. The 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 additional capabilities of video coding devices relative to existing devices according to, e.g., ITU-T H.264/AVC. For example, whereas H.264 provides nine intra-prediction encoding modes, the HM may provide as many as thirty-five intra-prediction encoding modes.

[0049] In general, the working model of the HM describes that a video picture may be divided into a sequence of treeblocks or largest coding units (LCU) that include both luma and chroma samples. Syntax data within a bitstream may define a size for the LCU, which is a largest coding unit in terms of the number of pixels. A slice includes a number of consecutive coding tree units (CTUs). Each of the CTUs may comprise a coding tree block of luma samples, two corresponding coding tree blocks of chroma samples, and syntax structures used to code the samples of the coding tree blocks. In a monochrome picture or a picture that have three separate color planes, a CTU may comprise a single coding tree block and syntax structures used to code the samples of the coding tree block.

[0050] A video picture may be partitioned into one or more slices. Each treeblock may be split into coding units (CUs) according to a quadtree. In general, a quadtree data

structure includes one node per CU, with a root node corresponding to the treeblock. 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. A CU may comprise a coding block of luma samples and two corresponding coding blocks of chroma samples of a picture that has a luma sample array, a Cb sample array and a Cr sample array, and syntax structures used to code the samples of the coding blocks. In a monochrome picture or a picture that have three separate color planes, a CU may comprise a single coding block and syntax structures used to code the samples of the coding block. A coding block is an NxN block of samples.

[0051] 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, four sub-CUs of a leaf-CU will also be referred to as leaf-CUs even if 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.

[0052] A CU has a similar purpose as a macroblock of the H.264 standard, except that a CU does not have a size distinction. For example, a treeblock may be split into four child nodes (also referred to as sub-CUs), and each child node may in turn be a parent node and be split into another four child nodes. A final, unsplit child node, referred to as a leaf node of the quadtree, comprises a coding node, also referred to as a leaf-CU. Syntax data associated with a coded bitstream may define a maximum number of times a treeblock may be split, referred to as a maximum CU depth, and may also define a minimum size of the coding nodes. Accordingly, a bitstream may also define a smallest coding unit (SCU). This disclosure uses the term “block” to refer to any of a CU, PU, or TU, in the context of HEVC, or similar data structures in the context of other standards (e.g., macroblocks and sub-blocks thereof in H.264/AVC).

[0053] A CU includes a coding node and prediction units (PUs) and transform units (TUs) associated with the coding node. A size of the CU corresponds to a size of the coding node and must be square in shape. The size of the CU may range from 8x8 pixels up to the size of the treeblock with a maximum of 64x64 pixels or greater. Each CU may contain one or more PUs and one or more TUs.

[0054] In general, a PU represents a spatial area corresponding to all or a portion of the corresponding CU, and may include data for retrieving a reference sample for the PU. Moreover, a PU includes data related to prediction. For example, when the PU is intra-mode encoded, data for the PU may be included in a residual quadtree (RQT), which may include data describing an intra-prediction mode for a TU corresponding to the PU. As another example, when the PU is inter-mode encoded, the PU may include data defining one or more motion vectors for the PU. A prediction block may be a rectangular (i.e., square or non-square) block of samples on which the same prediction is applied. A PU of a CU may comprise a prediction block of luma samples, two corresponding prediction blocks of chroma samples of a picture, and syntax structures used to predict the prediction block samples. In a monochrome picture or a picture that have three separate color planes, a PU may comprise a single prediction block and syntax structures used to predict the prediction block samples.

[0055] TUs may include coefficients in the transform domain following application of a transform, e.g., a discrete cosine transform (DCT), an integer transform, a wavelet transform, or a conceptually similar transform to residual video data. The residual data may correspond to pixel differences between pixels of the unencoded picture and prediction values corresponding to the PUs. Video encoder 20 may form the TUs including the residual data for the CU, and then transform the TUs to produce transform coefficients for the CU. A transform block may be a rectangular block of samples on which the same transform is applied. A transform unit (TU) of a CU may comprise a transform block of luma samples, two corresponding transform blocks of chroma samples, and syntax structures used to transform the transform block samples. In a monochrome picture or a picture that have three separate color planes, a TU may comprise a single transform block and syntax structures used to transform the transform block samples.

[0056] Following transformation, video encoder 20 may perform quantization of the transform coefficients. Quantization generally refers to a process in which transform coefficients are quantized to possibly reduce the amount of data used to represent the coefficients, providing further compression. The quantization process may reduce the bit depth associated with some or all of the coefficients. For example, an n -bit value may be rounded down to an m -bit value during quantization, where n is greater than m .

[0057] After video encoder 20 quantizes a coefficient block, video encoder 20 may entropy encode syntax elements indicating the quantized transform coefficients. For

example, video encoder 20 may perform Context-Adaptive Binary Arithmetic Coding (CABAC) on the syntax elements indicating the quantized transform coefficients. For entropy coding, video encoder 20 scans the quantized transform coefficients according to a particular scan order (e.g., vertical scan, horizontal scan, or diagonal scan). Video encoder 20 may output the entropy-encoded syntax elements in a bitstream.

[0058] Hence, the bitstream may include a sequence of bits that forms 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. The associated data may include video parameter sets (VPSs), sequence parameter sets (SPSs), picture parameter sets (PPSs), 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.

[0059] Transforming and quantizing the residual block causes loss of information (e.g., inverse quantized and inverse transformed block is different than the original residual block). Accordingly, examples of video coding where the residual block is transformed and quantized are referred to as lossy coding. In some examples, video encoder 20 may skip the transform of the residual block, but quantizes the residual block. Such examples of video coding are referred to as transform-skip coding. Transform-skip coding may be one variation of lossy coding because the quantization causes of loss of information. To avoid confusion, lossy coding is used in this description to refer to video coding methods that include both transform and quantization, and transform-skip coding is used in this description to refer to video coding methods in which the transform is skipped, but quantization is still performed.

[0060] Video encoder 20 need not perform transform-skip coding or lossy coding in all cases. In some examples, video encoder 20 may perform lossless coding. In lossless coding (sometimes referred to as transquant bypass), video encoder 20 does not transform the residual block and does not quantize the residual block. In this example, the residual block, as reconstructed by video decoder 30, is identical to the residual block generated by video encoder 20, whereas for lossy coding and transform-skip coding, the residual block, as reconstructed by video decoder 30, may be slightly different than the residual block generated by video encoder 20.

[0061] In other words, when transform is applied, the transform converts the residual values of the residual block from a pixel domain to a transform domain. In some examples, for transform skip or transform bypass, the residual data includes residual

values from the difference between the predictive block and the current block without a transform applied to the residual values that converts the residual values from a pixel domain to a transform domain.

[0062] Video decoder 30, upon obtaining the coded video data, may perform a decoding pass generally reciprocal to the encoding pass described with respect to video encoder 20. For example, video decoder 30 may obtain an encoded video bitstream that represents video blocks of an encoded video slice and associated syntax elements from video encoder 20. Video decoder 30 may reconstruct the original, unencoded video sequence (e.g., or a quantized version of the original residue for lossy coding) using the data contained in the bitstream. For example, for lossy coding, video decoder 30 may entropy decode syntax elements in the bitstream to determine quantized transform coefficients. Video decoder 30 may utilize the same scan order that video encoder 20 utilized to construct a quantized residual block. Video decoder 30 may then inverse quantize the quantized transform coefficients to determine the transform coefficients. Video decoder 30 may also apply an inverse transform to the transform coefficients to determine the coefficients of the residual block.

[0063] For transform-skip coding, video decoder 30 may entropy decode syntax elements in the bitstream to determine quantized coefficients, utilize the same scan order that video encoder 20 utilized to construct a quantized residual block, and then inverse quantize the quantized coefficients to determine the coefficients of the residual block. No inverse transform is needed because transform was skipped in the encoding process.

[0064] For lossless coding (e.g., transform bypass or simply bypass), video decoder 30 may entropy decode syntax elements in the bitstream and may utilize the same scan order video encoder 20 used to directly determine the coefficients of the residual block. No inverse quantizing or transform is needed because both transform and quantization was skipped in the encoding process.

[0065] In any case, video decoder 30 determines a predictive block. The predictive block may be located in the same picture (e.g., for intra-prediction or for intra-BC prediction) or in a different picture (e.g., for inter-prediction) as the current block. Video decoder 30 uses reconstructed pixel values in the predictive block and the corresponding residual values in the residual block (e.g., as obtained from the encoded bitstream) to reconstruct pixel values of the current block (e.g., to decode the current block).

[0066] In some instances, video encoder 20 and video decoder 30 may predict one or more residual values of a block using one or more other residual values from the block. For example, video encoder 20 may divide a residual block into one or more TUs. A “residual block” may generically refer to a block whose coefficients are to be entropy encoded (e.g., by video encoder 20) or entropy decoded (e.g., by video decoder 30). For instance, if video encoder 20 does not further divide the block resulting from the difference between the current block and a predictive block to generate a single TU, then the residual block corresponds the TU. If video encoder 20 divides the block resulting from the difference between the current block and a predictive block to generate a plurality of TUs, then the residual block corresponds to one of the plurality of TUs.

[0067] For inter-predicted blocks, video encoder 20 may explicitly signal whether RDPCM is on or off (e.g., whether RDPCM is applied or not) and direction of the RDPCM (if applied). Video decoder 30 may obtain the RDPCM syntax elements from the bitstream and apply RDPCM accordingly.

[0068] In some examples, video encoder 20 may provide the signaling at the TU level. In these examples, the block resulting from the difference between the predictive block and the original block is divided into TUs. Video encoder 20 then signals information (e.g., syntax elements such as a flag) indicating whether RDPCM is applied or not and information of the direction of the RDPCM (if applied) for each TU. In other examples, rather than at the TU level, video encoder 20 may signal the information indicating whether RDPCM is applied or not and the direction (e.g., mode/direction information of RDPCM) at CU or PU level. In these examples, the same mode/direction would be applicable to all TUs.

[0069] For intra-predicted blocks, video encoder 20 and video decoder 30 may apply RDPCM in the same manner, e.g., according to an intra-prediction direction. For example, for a horizontal intra-prediction mode, video encoder 20 may apply horizontal RDPCM, and for a vertical intra-prediction mode, video encoder 20 may apply vertical RDPCM. In some instances, video encoder 20 and video decoder 30 may not apply RDPCM for directional intra-prediction modes other than vertical and horizontal.

[0070] If RDPCM is applied for an intra-predicted block, video encoder 20 may not signal the RDPCM mode (e.g., RDPCM direction) in the bitstream. Rather, video encoder 20 may signal information indicating the intra-prediction mode, and video decoder 30 may determine that the RDPCM direction is the same as the intra-prediction

mode, in this example. In this way, for TUs (e.g., residual blocks) generated from intra-prediction, the application of RDPCM may be implicit, in that video encoder 20 may not signal information indicating the RDPCM direction. Rather, video decoder 30 simply determines the RDPCM mode (e.g., RDPCM direction) based on the intra-mode that is signaled for intra-predicted blocks.

[0071] Hence, while explicit and implicit signaling mechanisms are provided for applying RDPCM to inter-predicted data and intra-predicted data, respectively, RDPCM was not previously applied to intra-BC predicted residual. Accordingly, was not previously signaling mechanisms to indicate whether (and how) to apply RDPCM to intra-BC predicted residual.

[0072] According to aspects of this disclosure, video encoder 20 and/or video decoder 30 may apply intra-BC in conjunction with RDPCM. For example, video encoder 20 may generate a residual block for a current block of a picture based on a difference between the current block and a prediction block of the picture. Video encoder 20 may then generate a predicted residual block based on the residual block by predicting one or more residual values of the residual block based on one or more other residual values of the residual block. Video encoder 20 may then encode data that represents the predicted residual block in a bitstream.

[0073] For example, video encoder 20 may perform an intra-BC process to generate the residual block. As noted above, intra-BC may be a dedicated process that removes redundancy within a picture. As described in greater detail with respect to FIG. 4 below, for example, video encoder 20 may obtain a residual block for a CU being coded from an already reconstructed region in the same picture. In some instances, video encoder 20 may determine an offset vector that identifies the predictive block in the reconstructed region and may encode the offset vector (also referred to as a displacement vector or motion vector) together with the residue signal. In this way, the offset vector indicates the position of the predictive block in the picture used to generate the residual block as displaced from the current CU.

[0074] According to aspects of this disclosure, video encoder 20 may then generate the predicted residual block based on the intra-BC residual block by predicting one or more residual values of the residual block based on one or more other residual values of the residual block. For example, video encoder 20 may perform an RDPCM process on the residual block to generate the prediction block. Video encoder 20 may apply RDPCM by selecting one of three different RDPCM modes, including DPCM-off, RDPCM-

horizontal and RDPCM-vertical. For RDPCM-off, video encoder 20 may not apply RDPCM. For RDPCM horizontal, video encoder 20 may use pixel positions to the left of the pixel being coded, e.g., reconstructed residual values in the left column to predict the current residual value. For RDPCM vertical, video encoder 20 may use pixel positions above the pixel being coded, e.g., reconstructed residual values in the upper row to predict the current residual value. Accordingly, after applying RDPCM, the block may include predicted residual values of the original residual block.

[0075] According to aspects of this disclosure, video decoder 30 may generate a residual block of a picture based on a predicted residual block, where determining the residual block includes reconstructing one or more residual values of the residual block based on one or more predicted residual values of the residual block. Video decoder 30 then generates a current block of the picture based on a combination of the residual block and a prediction block of the picture.

[0076] For example, video decoder 30 may use an RDPCM process to generate the residual block of the picture based on the predicted residual block. As noted above with respect to video encoder 20, video decoder 30 may apply RDPCM by selecting one of three different RDPCM modes, including DPCM-off, RDPCM-horizontal and RDPCM-vertical. Video decoder 30 may obtain one or more syntax elements from an encoded bitstream that indicate the RDPCM mode.

[0077] Video decoder 30 may then perform an intra-BC prediction process to generate the current block of the picture. For example, video decoder 30 may obtain an offset vector from an encoded bitstream that indicates a location of a prediction block that is included in an already reconstructed portion of the picture being decoded. Video decoder may then combine the prediction block with the determined residual to generate (reconstruct) the current block.

[0078] The techniques of this disclosure also relate to signaling associated with applying RDPCM to residual that has been generated using an intra-BC mode. According to aspects of this disclosure, video encoder 20 may encode an indication of whether to apply RDPCM to blocks predicted using intra-BC. In some examples, video encoder 20 may encode one or more syntax elements for indicating whether to apply RDPCM in a manner similar to the explicit signaling described above with respect to inter-predicted data. For example, when predicting blocks using intra-BC, video encoder 20 may explicitly signal whether RDPCM is on or off (e.g., whether RDPCM is

applied or not) and a direction of the RDPCM (if applied). Video decoder 30 may obtain the RDPCM syntax elements from the bitstream and apply RDPCM accordingly. [0079] According to aspects of this disclosure, RDPCM may be enabled (or disabled) at a sequence level, e.g., according to one or more syntax elements included in an SPS. For example, video encoder 20 may encode one or more syntax elements indicating whether RDPCM is enabled. In some examples, the one or more syntax elements may be applicable to more than one prediction mode (e.g., where prediction modes include inter-prediction mode, intra-prediction mode, and intra-BC prediction mode). For example, video encoder 20 may encode a residual_dpcm_inter_enabled_flag syntax element (or a similar syntax element) to indicate whether RDPCM is enabled for both inter-predicted residual and intra-BC predicted residual (e.g., the flag is applicable to both modes such that one flag is signaled for both inter-prediction mode and intra-BC prediction mode). Video decoder 30 may decode such a flag from the bitstream and determine whether RDPCM is enabled. When enabled, one or more additional syntax elements may be explicitly signaled to indicate whether RDPCM is on or off for a block (e.g., whether RDPCM is applied or not) and a direction of the RDPCM for the block (if applied).

[0080] FIG. 2 is a block diagram illustrating an example of a video encoder 20 that may use techniques for transformation 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.

[0081] 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 picture. Inter-coding relies on temporal prediction to reduce or remove temporal redundancy in video within adjacent pictures of a video sequence. Intra-mode (I mode) may refer to any of several spatial based compression modes. Inter-modes, such as uni-directional prediction (P mode) or bi-prediction (B mode), may refer to any of several temporal-based compression modes.

[0082] In the example of FIG. 2, video encoder 20 includes video data memory 38, mode select unit 40, reference picture memory 64, summer 50, transform processing unit 52, quantization unit 54, and entropy encoding unit 56. Mode select unit 40, in turn, includes motion compensation unit 44, motion estimation unit 42, intra-prediction unit 46, intra-BC unit 47, and partition unit 48. For video block reconstruction, video

encoder 20 also includes inverse quantization unit 58, inverse transform unit 60, summer 62, and filtering unit 66.

[0083] Video data memory 38 may store video data to be encoded by the components of video encoder 20. The video data stored in video data memory 38 may be obtained, for example, from video source 18. Reference picture memory 64 may be referred to as a decoded picture buffer that stores reference video data for use in encoding video data by video encoder 20, e.g., in intra- or inter-coding modes. Video data memory 38 and reference picture memory 64 may be formed by any of a variety of memory devices, such as dynamic random access memory (DRAM), including synchronous DRAM (SDRAM), magnetoresistive RAM (MRAM), resistive RAM (RRAM), or other types of memory devices. Video data memory 38 and reference picture memory 64 may be provided by the same memory device or separate memory devices. In various examples, video data memory 38 may be on-chip with other components of video encoder 20, or off-chip relative to those components.

[0084] During the encoding process, video encoder 20 receives a video picture or slice to be coded. The picture or slice may be divided into multiple video blocks. Motion estimation unit 42 and motion compensation unit 44 perform inter-predictive coding of the received video block relative to one or more blocks in one or more reference pictures to provide temporal compression. Intra-prediction unit 46 may alternatively perform intra-predictive coding of the received video block relative to one or more neighboring pixel values in the same picture or slice as the block to be coded to provide spatial compression. Video encoder 20 may perform multiple coding passes, e.g., to select an appropriate coding mode for each block of video data.

[0085] Moreover, partition unit 48 may partition blocks of video data into sub-blocks, based on evaluation of previous partitioning schemes in previous coding passes. For example, partition unit 48 may initially partition a picture or slice into LCUs, and partition each of the LCUs into sub-CUs based on rate-distortion analysis (e.g., rate-distortion optimization). Mode select unit 40 may further produce a quadtree data structure indicative of partitioning of an LCU into sub-CUs. Leaf-node CUs of the quadtree may include one or more PUs and one or more TUs.

[0086] Mode select unit 40 may select one of the coding modes, intra or inter, e.g., based on error results, and provides the resulting intra- or inter-coded block to summer 50 to generate residual block data and to summer 62 to reconstruct the encoded block for use as a reference picture. Mode select unit 40 also provides syntax elements, such

as motion vectors, intra-mode indicators, partition information, and other such syntax information, to entropy encoding unit 56.

[0087] Motion estimation unit 42 and motion compensation unit 44 may be highly integrated, but are illustrated separately for conceptual purposes. Motion estimation, performed by motion estimation unit 42, is the process of generating motion vectors, which estimate motion for video blocks. A motion vector, for example, may indicate the displacement of a PU of a video block within a current video picture relative to a predictive block within a reference picture (or other coded unit) relative to the current block being coded within the current picture (or other coded unit).

[0088] A predictive block is a block that is found to closely match the block to be coded, in terms of pixel difference, which may be determined by sum of absolute difference (SAD), sum of square difference (SSD), or other difference metrics. In some examples, video encoder 20 may calculate values for sub-integer pixel positions of reference pictures stored in reference picture memory 64. For example, video encoder 20 may interpolate values of one-quarter pixel positions, one-eighth pixel positions, or other fractional pixel positions of the reference picture. Therefore, motion estimation unit 42 may perform a motion search relative to the full pixel positions and fractional pixel positions and output a motion vector with fractional pixel precision.

[0089] Motion estimation unit 42 calculates a motion vector for a PU of a video block in an inter-coded slice by comparing the position of the PU to the position of a predictive block of a reference picture. The reference picture may be selected from a first reference picture list (List 0) or a second reference picture list (List 1), each of which identify one or more reference pictures stored in reference picture memory 64. Motion estimation unit 42 sends the calculated motion vector to entropy encoding unit 56 and motion compensation unit 44.

[0090] Motion compensation, performed by motion compensation unit 44, may involve fetching or generating the predictive block based on the motion vector determined by motion estimation unit 42. Again, motion estimation unit 42 and motion compensation unit 44 may be functionally integrated, in some examples. Upon receiving the motion vector for the PU of the current video block, motion compensation unit 44 may locate the predictive block to which the motion vector points in one of the reference picture lists. Summer 50 forms a residual video block by subtracting pixel values of the predictive block from the pixel values of the current video block being coded, forming pixel difference values, as discussed below. In general, motion estimation unit 42

performs motion estimation relative to luma components, and motion compensation unit 44 uses motion vectors calculated based on the luma components for both chroma components and luma components. Mode select unit 40 may also generate syntax elements associated with the video blocks and the video slice for use by video decoder 30 in decoding the video blocks of the video slice.

[0091] Intra-prediction unit 46 may intra-predict a current block, as an alternative to the inter-prediction performed by motion estimation unit 42 and motion compensation unit 44, as described above. In particular, intra-prediction unit 46 may determine an intra-prediction mode to use to encode a current block. In some examples, intra-prediction unit 46 may encode a current block using various intra-prediction modes, e.g., during separate encoding passes, and intra-prediction unit 46 (or mode select unit 40, in some examples) may select an appropriate intra-prediction mode to use from the tested modes.

[0092] For example, intra-prediction unit 46 may calculate rate-distortion values using a rate-distortion analysis for the various tested intra-prediction modes, and select the intra-prediction mode having the best rate-distortion characteristics among the tested modes. Rate-distortion analysis generally determines an amount of distortion (or error) between an encoded block and an original, unencoded block that was encoded to produce the encoded block, as well as a bitrate (that is, a number of bits) used to produce the encoded block. Intra-prediction unit 46 may calculate ratios from the distortions and rates for the various encoded blocks to determine which intra-prediction mode exhibits the best rate-distortion value for the block.

[0093] Video encoder 20 forms a residual video block by subtracting the prediction data from mode select unit 40 from the original video block being coded. Summer 50 represents the component or components that perform this subtraction operation.

[0094] Transform processing unit 52 applies a transform, such as a discrete cosine transform (DCT) or a conceptually similar transform, to the residual block, producing a video block comprising residual transform coefficient values. Transform processing unit 52 may perform other transforms which are conceptually similar to DCT. Wavelet transforms, integer transforms, sub-band transforms or other types of transforms could also be used. In any case, transform processing unit 52 applies the transform to the residual block, producing a block of residual transform coefficients. The transform may convert the residual information from a pixel value domain to a transform domain, such as a frequency domain.

[0095] Transform processing unit 52 may send the resulting transform coefficients to quantization unit 54. Quantization unit 54 quantizes the transform coefficients to further reduce bit rate. The quantization process may reduce the bit depth associated with some or all of the coefficients. The degree of quantization may be modified by adjusting a quantization parameter. In some examples, quantization unit 54 may then perform a scan of the matrix including the quantized transform coefficients.

Alternatively, entropy encoding unit 56 may perform the scan.

[0096] Following quantization, entropy encoding unit 56 entropy codes the quantized transform coefficients. For example, entropy encoding unit 56 may perform context adaptive variable length coding (CAVLC), context adaptive binary arithmetic coding (CABAC), syntax-based context-adaptive binary arithmetic coding (SBAC), probability interval partitioning entropy (PIPE) coding or another entropy coding technique. In the case of context-based entropy coding, context may be based on neighboring blocks. Following the entropy coding by entropy encoding unit 56, the encoded bitstream may be transmitted to another device (e.g., video decoder 30) or archived for later transmission or retrieval.

[0097] Inverse quantization unit 58 and 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.

[0098] Motion compensation unit 44 may calculate a reference block by adding the residual block to a predictive block of one of the pictures of reference picture memory 64. 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. Summer 62 adds the reconstructed residual block to the motion compensated prediction block produced by motion compensation unit 44 to produce a reconstructed video block for storage in reference picture memory 64. The reconstructed video block may be used by motion estimation unit 42 and motion compensation unit 44 as a reference block to inter-code a block in a subsequent video picture.

[0099] Filtering unit 66 may perform a variety of filtering processes. For example, filtering unit 66 may perform deblocking. That is, filtering unit 66 may receive a plurality of reconstructed video blocks forming a slice or a frame of reconstructed video and filter block boundaries to remove blockiness artifacts from a slice or frame. In one example, filtering unit 66 evaluates the so-called “boundary strength” of a video block.

Based on the boundary strength of a video block, edge pixels of a video block may be filtered with respect to edge pixels of an adjacent video block such that the transition from one video block are more difficult for a viewer to perceive.

[0100] While the example of FIG. 2 generally illustrates video encoder 20 as a video encoder for performing lossy coding, the techniques of this disclosure may also be applied to lossless video coding. In some examples, lossless coding eliminates transforms and quantization. In other examples, lossless coding performs transforms and eliminates only the quantization process. In still other examples, lossless coding may be implemented with the use of transforms and quantitation, but the quantization parameter may be selected so as to avoid any quantization data loss. These and other examples are within the scope of this disclosure. In such instances, video encoder 20 may include components for performing transform skipping, quantization skipping, or other lossless coding techniques.

[0101] In accordance with various aspects of the techniques described in this disclosure, video encoder 20 may generate a residual block for a current block of a picture based on a difference between the current block and a prediction block of the picture. For example, intra-BC unit 47 may apply an intra-BC process to generate the residual block (as illustrated and described, for example, with respect to FIG. 4).

[0102] In addition, according to aspects of this disclosure, video encoder 20 generate a predicted residual block based on the residual block by predicting one or more residual values of the residual block based on one or more other values of the residual block. For example, intra-BC unit 47 may apply RDPCM to further compress the residual block. Video encoder 20 may then encode the predicted residual block, for example, by quantizing the predicted residual block using quantization unit 54, and entropy coding the quantized values.

[0103] While a number of different aspects and examples of the techniques are described in this disclosure, the various aspects and examples of the techniques may be performed together or separately from one another. In other words, the techniques should not be limited strictly to the various aspects and examples described above, but may be used in combination or performed together and/or separately. In addition, while certain techniques may be ascribed to certain units of video encoder 20 (such as intra-BC unit 47, motion compensation unit 44, or entropy encoding unit 56) it should be understood that one or more other units of video encoder 20 may also be responsible for carrying out such techniques.

[0104] FIG. 3 is a block diagram illustrating an example of video decoder 30 that may implement techniques for transformation as described in this disclosure. Again, the video decoder 30 will be described in the context of HEVC coding for purposes of illustration, but without limitation of this disclosure as to other coding standards.

[0105] In the example of FIG. 3, video decoder 30 includes video data memory 68, an entropy decoding unit 70, motion compensation unit 72, intra-prediction unit 74, intra-BC unit 75, inverse quantization unit 76, inverse transformation unit 78, reference picture memory 82, summer 80, and filtering unit 84.

[0106] Video data memory 68 may store video data, such as an encoded video bitstream, to be decoded by the components of video decoder 30. The video data stored in video data memory 68 may be obtained, for example, from computer-readable medium 16, e.g., from a local video source, such as a camera, via wired or wireless network communication of video data, or by accessing physical data storage media. Video data memory 68 may form a coded picture buffer (CPB) that stores encoded video data from an encoded video bitstream. Reference picture memory 82 may be referred to as a decoded picture buffer that stores reference video data for use in decoding video data by video decoder 30, e.g., in intra- or inter-coding modes. Video data memory 68 and reference picture memory 82 may be formed by any of a variety of memory devices, such as dynamic random access memory (DRAM), including synchronous DRAM (SDRAM), magnetoresistive RAM (MRAM), resistive RAM (RRAM), or other types of memory devices. Video data memory 68 and reference picture memory 82 may be provided by the same memory device or separate memory devices. In various examples, video data memory 68 may be on-chip with other components of video decoder 30, or off-chip relative to those components.

[0107] During the decoding process, video decoder 30 receives an encoded video bitstream that represents video blocks of an encoded video slice and associated syntax elements from video encoder 20. Entropy decoding unit 70 of video decoder 30 entropy decodes the bitstream to generate quantized coefficients, motion vectors or intra-prediction mode indicators, and other syntax elements. Entropy decoding unit 70 forwards the motion vectors to and other syntax elements to motion compensation unit 72. Video decoder 30 may receive the syntax elements at the video slice level and/or the video block level.

[0108] When the video slice is coded as an intra-coded (I) slice, intra-prediction unit 74 may generate prediction data for a video block of the current video slice based on a

signaled intra prediction mode and data from previously decoded blocks of the current picture. When the video picture is coded as an inter-coded (i.e., B, P or GPB) slice, motion compensation unit 72 produces predictive blocks for a video block of the current video slice based on the motion vectors and other syntax elements received from entropy decoding unit 70. The predictive blocks may be produced from one of the reference pictures within one of the reference picture lists. Video decoder 30 may construct the reference picture lists, List 0 and List 1, using default construction techniques based on reference pictures stored in reference picture memory 82.

[0109] Motion compensation unit 72 determines prediction information for a video block of the current video slice by parsing the motion vectors and other syntax elements, and uses the prediction information to produce the predictive blocks for the current video block being decoded. For example, motion compensation unit 72 uses some of the received syntax elements to determine a prediction mode (e.g., intra- or inter-prediction) used to code the video blocks of the video slice, an inter-prediction slice type (e.g., B slice, P slice, or GPB slice), construction information for one or more of the reference picture lists for the slice, motion vectors for each inter-encoded video block of the slice, inter-prediction status for each inter-coded video block of the slice, and other information to decode the video blocks in the current video slice.

[0110] Motion compensation unit 72 may also perform interpolation based on interpolation filters. Motion compensation unit 72 may use interpolation filters as used by video encoder 20 during encoding of the video blocks to calculate interpolated values for sub-integer pixels of reference blocks. In this case, motion compensation unit 72 may determine the interpolation filters used by video encoder 20 from the received syntax elements and use the interpolation filters to produce predictive blocks.

[0111] Inverse quantization unit 76 inverse quantizes, i.e., de-quantizes, the quantized transform coefficients provided in the bitstream and decoded by entropy decoding unit 70. The inverse quantization process may include use of a quantization parameter QP_Y calculated by video decoder 30 for each video block in the video slice to determine a degree of quantization and, likewise, a degree of inverse quantization that should be applied.

[0112] Inverse transform unit 78 applies an inverse transform, e.g., an inverse DCT, an inverse integer transform, or a conceptually similar inverse transform process, to the transform coefficients in order to produce residual blocks in the pixel domain. Video decoder 30 forms a decoded video block by summing the residual blocks from inverse

transform unit 78 with the corresponding predictive blocks generated by motion compensation unit 72. Summer 80 represents the component or components that perform this summation operation.

[0113] Filtering unit 84 may, in some examples, be configured similarly to filtering unit 66 of video encoder 20 (FIG. 2). For example, filtering unit 84 may be configured to perform deblocking, SAO, or other filtering operations when decoding and reconstructing video data from an encoded bitstream.

[0114] Again, as noted above with respect to FIG. 2, while the example of FIG. 3 generally illustrates video decoder 30 as a video encoder for performing lossy coding, the techniques of this disclosure may also be applied to lossless video coding. In some examples, lossless coding eliminates transforms and quantization. In other examples, lossless coding performs transforms and eliminates only the quantization process. In still other examples, lossless coding may be implemented with the use of transforms and quantitation, but the quantization parameter may be selected so as to avoid any quantization data loss. These and other examples are within the scope of this disclosure. In such instances, video decoder 30 may include components for performing transform skipping, quantization skipping, or other lossless coding techniques.

[0115] In some instances, video decoder 30 may perform one or more of the techniques described with respect to video encoder 20 above. For example, video decoder 30 may perform the techniques of this disclosure for applying both an intra-BC process and an RDPCM process to determine video data. In some examples, video decoder 30 may obtain one or more syntax elements from an encoded bitstream, and perform the techniques of this disclosure based on such syntax elements.

[0116] For example, in accordance with various aspects of the techniques described in this disclosure, video decoder 30 may perform RDPCM on received predicted residual to generate a residual block of a picture. That is, video decoder 30 may perform an inverse RDPCM process as that applied by video encoder 20.

[0117] In addition, video decoder 30, e.g., intra-BC unit 75 of video decoder 30, may determine a current block of the picture by combining the residual block of the picture and a block of the identified by an offset vector.

[0118] While a number of different aspects and examples of the techniques are described in this disclosure, the various aspects and examples of the techniques may be performed together or separately from one another. In other words, the techniques should not be limited strictly to the various aspects and examples described above, but

may be used in combination or performed together and/or separately. In addition, while certain techniques may be ascribed to certain units of video decoder 30 it should be understood that one or more other units of video decoder 30 may also be responsible for carrying out such techniques.

[0119] FIG. 4 is a diagram illustrating an example of an intra block copying process. The example of FIG. 4 includes a current coding unit (CU) 90, a prediction block 92 located in search region 94 and an offset vector 96. During encoding, video encoder 20 may encode residual for current CU 90 based on a difference between current CU 90 and prediction block 92 (which may also be referred to as a prediction signal). Video encoder 20 may locate prediction block 92 in search region 94, which has already been reconstructed in the same picture as current CU 90. Video encoder 20 may locate prediction block 92 using offset vector 96 (which may also be referred to as a “displacement vector”).

[0120] Video encoder 20 may encode offset vector 96 together with the residue signal. For example, video encoder 20 may include one or more syntax elements that identify or define a horizontal displacement component of offset vector 96 and a vertical displacement component of offset vector in an encoded video bitstream. Video encoder 20 may also encode the residue, e.g., the difference between pixel values of current CU 90 and pixel values of prediction block 92. Video decoder 30 may decode the one or more syntax elements to determine offset vector 96, and use the determined vector to identify prediction block 92 for current CU 90. Video decoder 30 may also decode the residue. Video decoder 30 may reconstruct current CU 90 by combining the decoded residue with reconstructed pixel values of prediction block 92 (as identified by offset vector 96).

[0121] In some examples, the resolution of offset vector 96 may be integer pixel, e.g., be constrained to have integer pixel resolution. In such examples, the resolution of the horizontal displacement component and the vertical displacement component are integer pixel. In such examples, video encoder 20 and video decoder 30 need not interpolate pixel values of prediction block 92 to determine the predictor for current CU 90. In other examples, the resolution of one or both of the horizontal displacement component and the vertical displacement component may be sub-pixel in resolution. For example, one of the vertical and horizontal components may have integer pixel resolution, while the other has sub-pixel resolution.

[0122] According to aspects of this disclosure, video encoder 20 and/or video decoder 30 may code (encode or decode, respectively) blocks of video data using the intra-BC techniques shown in the example of FIG. 4. In addition, as described in greater detail below with respect to FIGS. 5A and 5B, video encoder 20 and/or video decoder may apply RDPCM to the residue that has been generated using the intra-BC techniques.

[0123] For example, as noted above, video encoder 20 may determine offset vector 96, which identifies a location of a block (i.e., prediction block 92) that closely matches current CU 90. Video encoder 20 may determine a residue based on a difference between current CU 90 and prediction block 92. According to aspects of this disclosure, video encoder 20 may then apply RDPCM to the determined residual. Video encoder 20 may encode the residual values that have been predicted using RDPCM, as well as offset vector 96.

[0124] Video decoder 30 may decode residue from an encoded bitstream. As noted above, the residue represents the difference between current CU 90 and prediction block 92. According to aspects of this disclosure, the decoded residue may include predicted residual values, e.g., residual values that have been predicted using RDPCM. Accordingly, video decoder 30 may apply RDPCM to reconstruct the residue (e.g., the difference between current CU 90 and prediction block 92). Video decoder 30 may then locate prediction block 92 using offset vector 96 and combine the reconstructed residue with prediction block 92 to determine current CU 90.

[0125] The techniques may be applied in both lossless and lossy coding schemes. For example, according to aspects of this disclosure, RDPCM may be applied to residue that has been generated using intra-BC, regardless of whether the residue subsequently subjected to quantization, as described above.

[0126] FIGS. 5A and 5B are diagrams illustrating examples of residue differential pulse code modulation (RDPCM) techniques. As noted above, two RDPCM modes include an RDPCM vertical mode and an RDPCM horizontal mode. The concept behind RDPCM is to use the upper row pixel to predict the current pixel for vertical mode and to use the left column pixel for predicting the current pixel for horizontal mode. For example, consider a residue block of size M (rows) \times N (cols). Let r_{ij} , $0 \leq i \leq (M - 1)$, $0 \leq j \leq (N - 1)$ be the residue (e.g., prediction residual after performing inter- intra- or intra-BC prediction). The residue block may represent any component of a video block (e.g. a luma component, a chroma component, a Red component, a Green component, a Blue component (in RGB coding) or the like).

[0127] In residual DPCM, prediction is applied to the residual samples, so that a modified $M \times N$ array \tilde{R} with elements $\tilde{r}_{i,j}$ is obtained as follows for lossless vertical residual DPCM:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & i = 0, 0 \leq j \leq (N-1) \\ r_{i,j} - r_{(i-1),j}, & 1 \leq i \leq (M-1), 0 \leq j \leq (N-1) \end{cases},$$

or for lossless horizontal RDPCM:

$$\tilde{r}_{i,j} = \begin{cases} r_{i,j}, & 0 \leq i \leq (M-1), j = 0 \\ r_{i,j} - r_{i,(j-1)}, & 0 \leq i \leq (M-1), 1 \leq j \leq (N-1) \end{cases}.$$

[0128] Video encoder 20 signals the modified residual samples of \tilde{R} instead of the original residual samples R in an encoded bitstream. As described herein, the modified residual samples may be referred to as predicted residual samples, because the modified residual samples are essentially predicted from adjacent residual samples.

[0129] In other words, if the residual block included samples R , then video encoder 20 may determine the modified residual samples \tilde{R} , which represents the residual block with RDPCM applied. Video encoder 20 may apply the same RDPCM process regardless of whether quantization is performed (e.g., the lossy case) or quantization is skipped (e.g. the lossless case). In some examples, video encoder 20 may extend RDPCM to the lossy case only for those TUs where the transform is skipped. In particular, for lossy RDPCM-vertical:

$$\tilde{r}(i,j) = \begin{cases} Q(r(i,j)) & i = 0 \text{ and } 0 \leq j \leq N-1 \\ Q(r(i,j) - \hat{r}(i-1,j)) & 0 < i \leq M-1 \text{ and } 0 \leq j \leq N-1 \end{cases}$$

or for RDPCM-horizontal:

$$\tilde{r}(i,j) = \begin{cases} Q(r(i,j)) & j = 0 \text{ and } 0 \leq i \leq N-1 \\ Q(r(i,j) - \hat{r}(i,j-1)) & 0 < i \leq M-1 \text{ and } 0 \leq j \leq N-1 \end{cases}$$

where Q denotes the forward quantisation operation and \hat{r} is a reconstructed version of residual values r .

[0130] At the decoder side, video decoder 30 may parse the modified (e.g., predicted) residual samples from an encoded bitstream. Video decoder 30 may then reconstruct the residual samples as follows for lossless RDPCM-vertical:

$$r_{i,j} = \sum_{k=0}^i \tilde{r}_{k,j}, \quad 0 \leq i \leq (M-1), 0 \leq j \leq (N-1),$$

or for lossless RDPCM-horizontal:

$$r_{i,j} = \sum_{k=0}^l \tilde{r}_{i,k}, \quad 0 \leq i \leq (M-1), \quad 0 \leq j \leq (N-1).$$

Video decoder 30 may also apply RDPCM in lossy and lossless video coding. For example, video decoder 30 may add the reconstructed residuals after inverse quantisation according to the selected RDPCM mode to obtain the reconstructed residuals $\hat{r}(i, j)$ for lossy RDPCM-vertical:

$$\hat{r}(i, j) = \sum_{k=0}^l Q^{-1}(\tilde{r}(k, j))$$

or for lossy RDPCM-horizontal:

$$\hat{r}(i, j) = \sum_{k=0}^l Q^{-1}(\tilde{r}(i, k))$$

where Q^{-1} denotes the inverse quantisation operation.

[0131] As noted above, for inter-RDPCM (e.g., applying RDPCM to motion compensated residue) video encoder 20 may explicitly signal whether to apply RDPCM to video decoder 30 in an encoded bitstream. Video encoder 20 may also explicitly signal one or more syntax elements in an encoded bitstream indicating the RDPCM mode (e.g., no RDPCM, vertical RDPCM, or horizontal RDPCM) for a particular block (or blocks) of video data. Video decoder 30 may receive the one or more syntax elements and perform the RDPCM in a reciprocal manner, as noted above.

[0132] According to aspects of this disclosure, video encoder 20 and/or video decoder may apply RDPCM to residue that has been generated using an intra-BC coding process. For example, as noted above, video encoder 20 may determine an offset vector that identifies a location of a prediction block that closely matches the block currently being coded. Video encoder 20 may determine a residue based on a difference between the current block and the prediction block.

[0133] Video encoder 20 may determine an RDPCM mode. In some examples, video encoder 20 may determine the RDPCM mode using a rate-distortion analysis (e.g., a rate-distortion optimization). Video encoder 20 may select between RDPCM off (e.g., not applying RDPCM) a vertical mode (as shown in the example of FIG. 5A) or a horizontal mode (e.g., as shown in the example of FIG. 5B). Video encoder 20 may use the determined mode to generate modified residual values, e.g., according to the process described above, which may also be referred to as predicted residual values. Video encoder 20 may then encode the predicted residual values, as well as the determined offset vector.

[0134] Video decoder 30 may decode residue from an encoded bitstream. The decoded residue may include modified residual values, e.g., residual values that have been predicted using RDPCM. Video decoder 30 may apply RDPCM to reconstruct the residue (e.g., a quantized version of the original residue). For example, video decoder 30 may apply a reciprocal RDPCM process as video encoder 20 to reconstruct an unmodified residue.

[0135] According to aspects of this disclosure, video decoder 30 may decode one or more syntax elements to determine the appropriate RDPCM mode to apply to the modified residue. For example, the one or more syntax elements may indicate whether to select RDPCM off (e.g., not applying RDPCM) a vertical mode (as shown in the example of FIG. 5A) or a horizontal mode (e.g., as shown in the example of FIG. 5B). As noted above, the one or more syntax elements may be provided at the CU, TU, or PU level.

[0136] After reconstructing the residue, video decoder 30 may locate the prediction block using an offset vector obtained from the encoded bitstream. Video decoder 30 may then reconstruct the block currently being decoded by combining the reconstructed residue (e.g., a quantized version of the original residue) with the prediction block. As noted above, the techniques may be applied in both lossless and lossy coding schemes. For example, according to aspects of this disclosure, RDPCM may be applied to residue that has been generated using intra-BC, regardless of whether the residue subsequently subjected to quantization.

[0137] According to aspects of this disclosure, RDPCM may be enabled (or disabled) at a sequence level, e.g., according to one or more syntax elements included in an SPS. For example, video encoder 20 may encode one or more syntax elements indicating whether RDPCM is enabled for intra-BC predicted data. In general, when enabled, video decoder 30 may expect to decode information indicating a particular RDPCM mode whenever video decoder 30 decodes a block that has been encoded with an intra-BC process.

[0138] In one example, a residual_dpcm_inter_enabled_flag syntax element may be used indicate whether RDPCM may be applied to a block that has been predicted using intra-BC. Flynn et al., “High Efficiency Video Coding (HEVC) Range Extensions text specification: Draft 4,” JCTVC-N1005_v3, August 2013, by D. Flynn, J. Sole, and T. Suzuki may provide some description related to the residual_dpcm_inter_enabled_flag at the SPS level. In some examples, when the residual_dpcm_inter_enabled_flag syntax

element is equal to one, RDPCM is available for use by video encoder 20 and video decoder 30 for blocks predicted using intra-BC. In such examples, when the residual_dpcm_inter_enabled_flag syntax element is equal to zero, RDPCM is not available for use by video encoder 20 and video decoder 30 for blocks predicted using intra-BC. It should be understood that the syntax element values described above are provided for purposes of illustration and should not be considered limiting. Also, these example syntax elements need not necessarily be part of the SPS.

[0139] In some examples, the one or more syntax elements that indicate whether RDPCM is enabled may be applicable to more than one prediction mode (e.g., where prediction modes include inter-prediction mode, intra-prediction mode, and intra-BC prediction mode). For example, video encoder 20 may encode a residual_dpcm_inter_enabled_flag syntax element (or a similar syntax element) to indicate whether RDPCM is enabled for both inter-predicted residual and intra-BC predicted residual (e.g., the flag is applicable to both modes such that one flag is signaled for inter-prediction mode and intra-BC prediction mode). Video decoder 30 may decode such a flag from the bitstream and determine whether RDPCM is enabled. When enabled, one or more additional syntax elements may be explicitly signaled to indicate whether RDPCM is on or off for a block (e.g., whether RDPCM is applied or not) and a direction of the RDPCM for the block (if applied).

[0140] The residual_dpcm_inter_enabled_flag syntax element described above is merely one example. In another example, instead of or in addition to the residual_dpcm_inter_enabled_flag syntax element, a residual_dpcm_intra_enabled_flag syntax element may be included at the SPS level to control whether the RDPCM is enabled for the blocks predicted with intra-BC. The JCTVC-N1005_v3 document may also provide description related to the residual_dpcm_intra_enabled_flag.

[0141] FIG. 6 is a flowchart illustrating exemplary operation of a video encoding device, such as video encoder 20 shown in the example of FIG. 2, in performing the techniques described in this disclosure. While described with respect to video encoder 20, it should be understood that the techniques of FIG. 6 may be carried out by a variety of other devices having a variety of other processing capabilities.

[0142] In the example of FIG. 6, video encoder 20 receives a video picture or slice to be coded (100). In performing the intra-BC process, intra-BC unit 47 may first identify a search region in the video picture for a current block (e.g., CU) of the video picture (102). After identifying the appropriate search region, intra-BC unit 47 may next

determine a prediction block within the search region from which the current block is to be predicted (104). Intra-BC unit 47 may perform this determination by accessing one or more blocks within the search region and determining the difference between each block and the current CU. Intra-BC unit 47 may determine the prediction block as the block that results in the least amount of residual or, in other words, smallest difference between the block and the current CU.

[0143] Intra-BC unit 47 may next determine an offset vector identifying the location of the selected block (which may be referred to as the “prediction block”) relative to the current CU. In this way, intra-BC unit 47 may determine the offset vector based on the prediction block and the current block (106). Intra-BC unit 47 may pass the offset vector to entropy encoding unit 56, which entropy encodes the offset vector and adds the encoded offset vector to the bitstream. Intra-BC unit 47 also determines a residual as the difference between the prediction block and the current block (108) and passes the residual as a residual block to the transform processing unit 52.

[0144] According to aspects of this disclosure, video encoder 20 may generate modified residual values for the residual block (110). For example, video encoder 20 may predict one or more residual values of the residual block from one or more other residual values of the residual block. In some examples, video encoder 20 may generate the modified residual values by applying RDPCM to the residual block. As described above with respect to FIGS. 5A and 5B, video encoder may apply a vertical RDPCM mode or a horizontal RDPCM mode to generate the modified residual values.

[0145] Quantization unit 54 quantizes the modified residual values to further reduce bit rate (112). While the example process shown in FIG. 6 includes quantization, in other examples, as noted above, quantization may be skipped (e.g., in a lossless coding process). In such instances, video encoder 20 may pass the modified residual values directly to entropy encoding unit 56.

[0146] Following quantization (if performed), entropy encoding unit 56 entropy codes (which refers to statistical lossless coding, as one example) the quantized values (114) and includes the entropy encoded quantized values in the bitstream. This process iterates until all of the blocks of the video picture are encoded, although not all of the block may be encoded using an intra-BC process.

[0147] While certain techniques may be ascribed to certain units of video encoder 20, it should be understood that one or more other units of video encoder 20 may also be responsible for carrying out such techniques.

[0148] FIG. 7 is a flowchart illustrating exemplary operation of a video decoding device, such as video decoder 30 shown in the example of FIG. 3, in performing the techniques described in this disclosure. Again, while described with respect to video decoder 30, it should be understood that the techniques of FIG. 7 may be carried out by a variety of other devices having a variety of other processing capabilities.

[0149] Initially, video decoder 30 receives an encoded video bitstream that represents video blocks of an encoded video slice and associated syntax elements from video encoder 20. Entropy decoding unit 70 of video decoder 30 entropy decodes the bitstream to generate quantized coefficients, and in the case of intra-BC, a corresponding an offset vector (122). While the process shown in the example of FIG. 7 is described with respect to a lossy coding process (with quantization applied to residue), as noted above, the techniques may also be performed in lossless video coding.

[0150] In the case of lossy coding, entropy decoding unit 70 forwards the quantized coefficients to inverse quantization unit 76 and the corresponding offset vector to intra-BC unit 75. Inverse quantization unit 76 dequantizes the quantized coefficients to obtain a modified residual block (124) (in instances in which quantization is applied). The modified residual block may also be referred to as a predicted residual block, because, as described below, values of an unmodified residual block to which the modified residual block corresponds have been predicted using other values of the unmodified residual block.

[0151] According to aspects of this disclosure, video decoder 30 may reconstruct a residual block based on the values of the modified residual block (126). For example, video decoder 30 may apply a reciprocal process as video encoder 20 to reconstruct the residual values of the block from the received modified residual values of the block. In some examples, video decoder 30 may reconstruct the residual block by applying RDPCM to the modified residual block. As described above with respect to FIGS. 5A and 5B, video decoder 30 may apply a vertical RDPCM mode or a horizontal RDPCM mode to reconstruct the residual values of the residual block.

[0152] Intra-BC unit 75 identifies a prediction block stored to reference picture memory 82 (or some other intermediate memory) based on the offset vector (128) and provides this prediction block to summer 80. Summer 80 sums the residual block with the prediction block to decode (e.g., reconstruct) the coded current block (130).

[0153] While certain techniques may be ascribed to certain units of video decoder 30, it should be understood that one or more other units of video decoder 30 may also be responsible for carrying out such techniques.

[0154] It is to be recognized that depending on the example, certain acts or events of any of the techniques described herein can be performed in a different sequence, may be added, merged, or left out altogether (e.g., not all described acts or events are necessary for the practice of the techniques). Moreover, in certain examples, acts or events may be performed concurrently, e.g., through multi-threaded processing, interrupt processing, or multiple processors, rather than sequentially.

[0155] Certain aspects of this disclosure have been described with respect to the developing HEVC standard for purposes of illustration. However, the techniques described in this disclosure may be useful for other video coding processes, including other standard or proprietary video coding processes not yet developed.

[0156] A video coder, as described in this disclosure, may refer to a video encoder or a video decoder. Similarly, a video coding unit may refer to a video encoder or a video decoder. Likewise, video coding may refer to video encoding or video decoding, as applicable.

[0157] 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.

[0158] 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.

[0159] 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.

[0160] 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.

[0161] 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.

[0162] 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.

[0163] Various examples have been described. These and other examples are within the scope of the following claims.

WHAT IS CLAIMED IS:

1. A method of decoding video data, the method comprising:
 - generating a residual block of a picture based on a predicted residual block including reconstructing one or more residual values of the residual block based on one or more predicted residual values of the residual block; and
 - generating a current block of the picture based on a combination of the residual block and a prediction block of the picture.
2. The method of claim 1, further comprising obtaining the predicted residual block from an encoded bitstream, and wherein generating the residual block comprises applying residual differential pulse code modulation (RDPCM) to the predicted residual block.
3. The method of claim 2, wherein applying the RDPCM to the predicted residual block comprises applying horizontal RDPCM to the one or more predicted residual values.
4. The method of claim 2, wherein applying the RDPCM to the predicted residual block comprises applying vertical RDPCM to the one or more predicted residual values.
5. The method of claim 2, wherein applying the RDPCM comprises obtaining, from the encoded bitstream, data that indicates an RDPCM mode from a plurality of RDPCM modes and applying the indicated RDPCM mode.
6. The method of claim 5, wherein obtaining the data that indicates an RDPCM mode comprises obtaining data that indicates at least one of an RDPCM off mode, an RDPCM vertical mode, and an RDPCM horizontal mode.
7. The method of claim 1, further comprising:
 - obtaining, from an encoded bitstream, a displacement vector that indicates a location of the prediction block in the picture; and
 - locating the prediction block of the picture using the displacement vector.

8. The method of claim 1, wherein generating the residual block further comprises applying inverse quantization to the residual block in a lossy decoding process.

9. The method of claim 1, wherein generating the residual block comprises generating the residual block without performing inverse quantization in a lossless decoding process.

10. The method of claim 1, further comprising decoding one or more syntax elements that indicate whether an RDPCM process is enabled, wherein the RDPCM process includes the step of generating the residual block of the picture based on the predicted residual block.

11. The method of claim 10, wherein decoding the one or more syntax elements comprises decoding the one or more syntax elements at a sequence level in a sequence parameter set (SPS).

12. The method of claim 10, wherein the one or more syntax elements that indicate whether the RDPCM process is enabled are applicable to an inter-prediction mode and an intra-BC prediction mode.

13. A method of encoding video data, the method comprising:
generating a residual block for a current block of a picture based on a difference between the current block and a prediction block of the picture;
generating a predicted residual block based on the residual block by predicting one or more residual values of the residual block based on one or more other residual values of the residual block; and
encoding data that represents the predicted residual block in a bitstream.

14. The method of claim 13, wherein generating the predicted residual block comprises applying residual differential pulse code modulation (RDPCM) to the residual block.

15. The method of claim 14, wherein applying the RDPCM to the residual block comprises applying horizontal RDPCM to the one or more residual values.

16. The method of claim 14, wherein applying the RDPCM to the predicted residual block comprises applying vertical RDPCM to the one or more residual values.
17. The method of claim 14, wherein applying the RDPCM comprises selecting an RDPCM mode from a plurality of RDPCM modes and applying the selected RDPCM mode, the method further comprising encoding data that indicates the selected RDPCM mode.
18. The method of claim 17, wherein encoding data that indicates the selected RDPCM mode comprises encoding data that indicates at least one of an RDPCM off mode, an RDPCM vertical mode, and an RDPCM horizontal mode.
19. The method of claim 13, wherein generating the residual block comprises applying an intra-block copying (intra-BC) process to generate the residual block, wherein applying the intra-BC process comprises determining a region of the picture from which to select the prediction block, and determining a displacement vector that identifies the prediction block, the method further comprising encoding data that represents the displacement vector.
20. The method of claim 13, wherein encoding the data comprises applying quantization to the predicted residual block in a lossy encoding process.
21. The method of claim 13, wherein encoding the data comprises encoding the data without applying quantization to the predicted residual in a lossless encoding process.
22. The method of claim 13, further comprising encoding one or more syntax elements that indicate whether an RDPCM process is enabled, wherein the RDPCM process includes the step of generating the predicted residual block.
23. The method of claim 22, wherein encoding the one or more syntax elements comprises encoding the one or more syntax elements at a sequence level in a sequence parameter set (SPS).

24. The method of claim 22, wherein the one or more syntax elements that indicate whether the RDPCM process is enabled are applicable to an intra-BC mode and an inter-prediction mode.

25. A device for decoding video data, the device comprising:
a memory configured to store the video data; and
a video decoder configured to:
generate a residual block of a picture based on a predicted residual block including reconstructing one or more residual values of the residual block based on one or more predicted residual values of the residual block; and
generate a current block of the picture based on a combination of the residual block and a prediction block of the picture.

26. The device of claim 25, wherein the video decoder is further configured to obtain the predicted residual block from an encoded bitstream, and wherein to generate the residual block, the video decoder is configured to apply residual differential pulse code modulation (RDPCM) to the predicted residual block.

27. The device of claim 26, wherein to apply the RDPCM to the predicted residual block, the video decoder is configured to apply horizontal RDPCM to the one or more predicted residual values.

28. The device of claim 26, wherein to apply the RDPCM to the predicted residual block, the video decoder is configured to apply vertical RDPCM to the one or more predicted residual values.

29. The device of claim 26, wherein to apply the RDPCM, the video decoder is configured to obtain, from the encoded bitstream, data that indicates an RDPCM mode from a plurality of RDPCM modes and applying the indicated RDPCM mode.

30. The device of claim 29, wherein to obtain the data that indicates an RDPCM mode, the video decoder is configured to obtain data that indicates at least one of an RDPCM off mode, an RDPCM vertical mode, and an RDPCM horizontal mode.

31. The device of claim 25, wherein the video decoder is further configured to:
obtain, from an encoded bitstream, a displacement vector that indicates a
location of the prediction block in the picture; and

locate the prediction block of the picture using the displacement vector.

32. The device of claim 25, wherein to generate the residual block, the video
decoder is configured to apply inverse quantization to the residual block in a lossy
decoding process.

33. The device of claim 25, wherein to generate the residual block, the video
decoder is configured to generate the residual block without performing inverse
quantization in a lossless decoding process.

34. The device of claim 25, wherein the video decoder is further configured to
decode one or more syntax elements that indicate whether an RDPCM process is
enabled, wherein the RDPCM process includes the generating the residual block of the
picture based on the predicted residual block.

35. The device of claim 34, wherein to decode the one or more syntax elements, the
video decoder is configured to decode the one or more syntax elements at a sequence
level in a sequence parameter set (SPS).

36. The device of claim 34, wherein the one or more syntax elements that indicate
whether the RDPCM process is enabled are applicable to an inter-prediction mode and
an intra-BC prediction mode.

37. The device of claim 25, further comprising a display configured to display the
current block of the picture.

38. A device for encoding video data, the device comprising:

a memory configured to store the video data;

a video encoder configured to:

generate a residual block for a current block of a picture based on a
difference between the current block and a prediction block of the picture;

generate a predicted residual block based on the residual block by predicting one or more residual values of the residual block based on one or more other residual values of the residual block; and encode data that represents the predicted residual block in a bitstream.

39. The device of claim 38, wherein to generate the predicted residual block, the video encoder is configured to apply residual differential pulse code modulation (RDPCM) to the residual block.

40. The device of claim 39, wherein to apply the RDPCM to the residual block, the video encoder is configured to apply horizontal RDPCM to the one or more residual values.

41. The device of claim 39, wherein to apply the RDPCM to the predicted residual block, the video encoder is configured to apply vertical RDPCM to the one or more residual values.

42. The device of claim 39, wherein to apply the RDPCM, the video encoder is configured to select an RDPCM mode from a plurality of RDPCM modes and to apply the selected RDPCM mode, the video encoder is further configured to encode data that indicates the selected RDPCM mode.

43. The device of claim 42, wherein to encode the data that indicates the selected RDPCM mode, the video encoder is configured to encode data that indicates at least one of an RDPCM off mode, an RDPCM vertical mode, and an RDPCM horizontal mode.

44. The device of claim 38, wherein to generate the residual block, the video encoder is configured to apply an intra-block copying (intra-BC) process to generate the residual block, wherein to apply the intra-BC process, the video encoder is configured to determine a region of the picture from which to select the prediction block and to determine a displacement vector that identifies the prediction block, and wherein the video encoder is further configured to encode data that represents the displacement vector.

45. The device of claim 38, wherein to encode the data, the video encoder is configured to apply quantization to the predicted residual block in a lossy encoding process.

46. The device of claim 38, wherein to encode the data, the video encoder is configured to encode the data without applying quantization to the predicted residual in a lossless encoding process.

47. The device of claim 38, wherein the video encoder is further configured to encode one or more syntax elements that indicate whether an RDPCM process is enabled, wherein the RDPCM process includes the generating the predicted residual block.

48. The device of claim 47, wherein to encode the one or more syntax elements, the video encoder is configured to encode the one or more syntax elements at a sequence level in a sequence parameter set (SPS).

49. The device of claim 47, wherein the one or more syntax elements that indicate whether the RDPCM process is enabled are applicable to an intra-BC mode and an inter-prediction mode.

50. The device of claim 38, further comprising a camera sensor configured to generate the current block.

51. A device for decoding video data, the device comprising:
means for generating a residual block of a picture based on a predicted residual block including reconstructing one or more residual values of the residual block based on one or more predicted residual values of the residual block; and
means for generating a current block of the picture based on a combination of the residual block and a prediction block of the picture.

52. The device of claim 51, further comprising means for decoding one or more syntax elements that indicate whether an RDPCM process is enabled, wherein the RDPCM process includes the generating the residual block of the picture based on the

predicted residual block, and wherein the one or more syntax elements that indicate whether the RDPCM process is enabled are applicable to an inter-prediction mode and an intra-BC prediction mode.

53. A device for encoding video data, the device comprising:

- means for generating a residual block for a current block of a picture based on a difference between the current block and a prediction block of the picture;
- means for generating a predicted residual block based on the residual block by predicting one or more residual values of the residual block based on one or more other residual values of the residual block; and
- means for encoding data that represents the predicted residual block in a bitstream.

54. The device of claim 53, further comprising means for encoding one or more syntax elements that indicate whether an RDPCM process is enabled, wherein the RDPCM process includes the generating the predicted residual block, and wherein the one or more syntax elements that indicate whether the RDPCM process is enabled are applicable to an intra-BC mode and an inter-prediction mode.

55. A non-transitory computer-readable medium having instructions stored thereon that, when executed, cause one or more processors to:

- generate a residual block of a picture based on a predicted residual block including reconstructing one or more residual values of the residual block based on one or more predicted residual values of the residual block; and
- generate a current block of the picture based on a combination of the residual block and a prediction block of the picture.

56. The non-transitory computer-readable medium of claim 55, wherein the instructions further cause the one or more processors to decode one or more syntax elements that indicate whether an RDPCM process is enabled, wherein the RDPCM process includes the generating the residual block of the picture based on the predicted residual block, and wherein the one or more syntax elements that indicate whether the RDPCM process is enabled are applicable to an inter-prediction mode and an intra-BC prediction mode.

57. A non-transitory computer-readable medium having instructions stored thereon that, when executed, cause one or more processors to:

generate a residual block for a current block of a picture based on a difference between the current block and a prediction block of the picture;

generate a predicted residual block based on the residual block by predicting one or more residual values of the residual block based on one or more other residual values of the residual block; and

encode data that represents the predicted residual block in a bitstream.

58. The non-transitory computer-readable medium of claim 57, wherein the instructions further cause the one or more processors to encode one or more syntax elements that indicate whether an RDPCM process is enabled, wherein the RDPCM process includes the generating the predicted residual block, and wherein the one or more syntax elements that indicate whether the RDPCM process is enabled are applicable to an intra-BC mode and an inter-prediction mode.

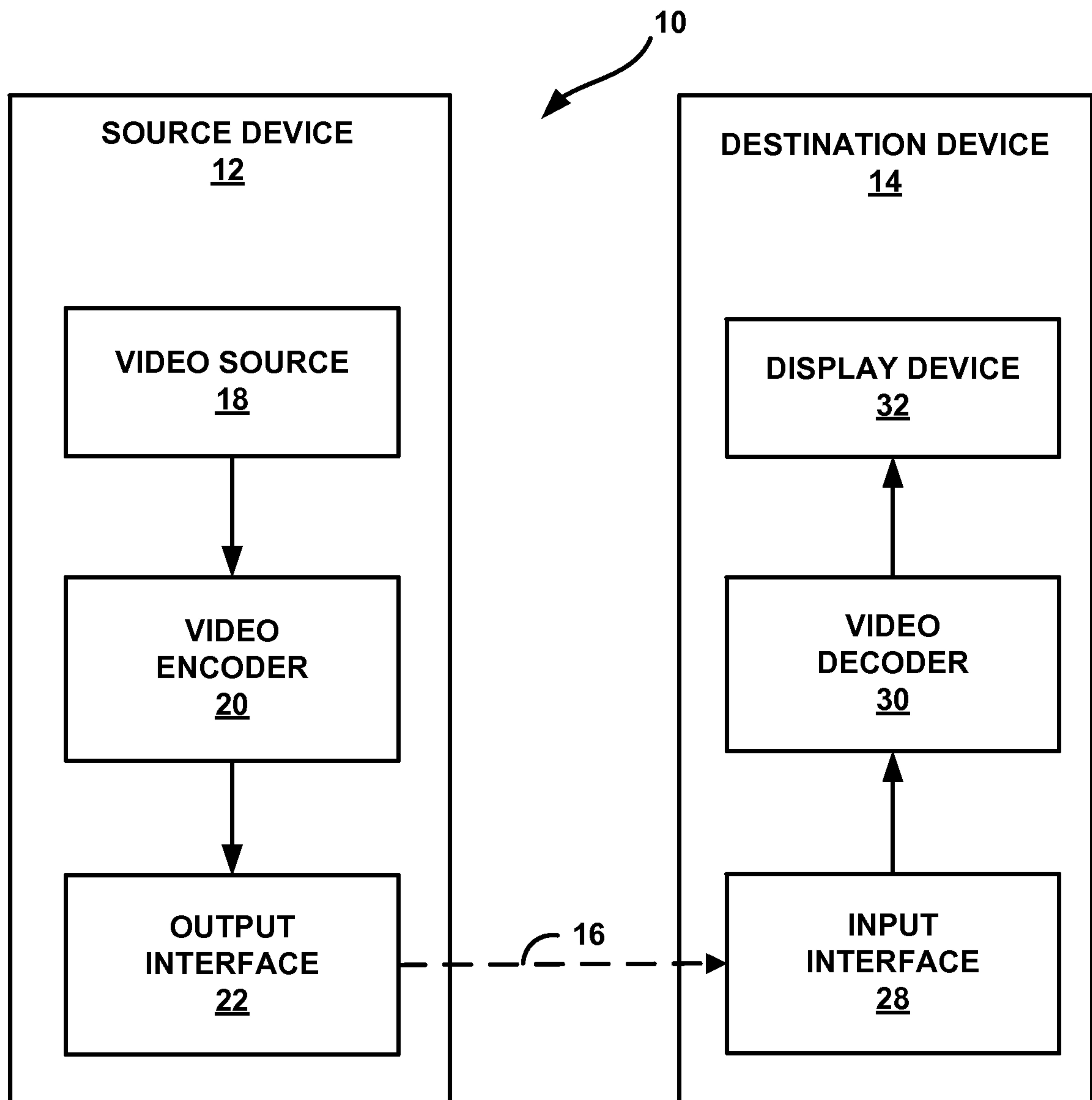


FIG. 1




FIG. 2

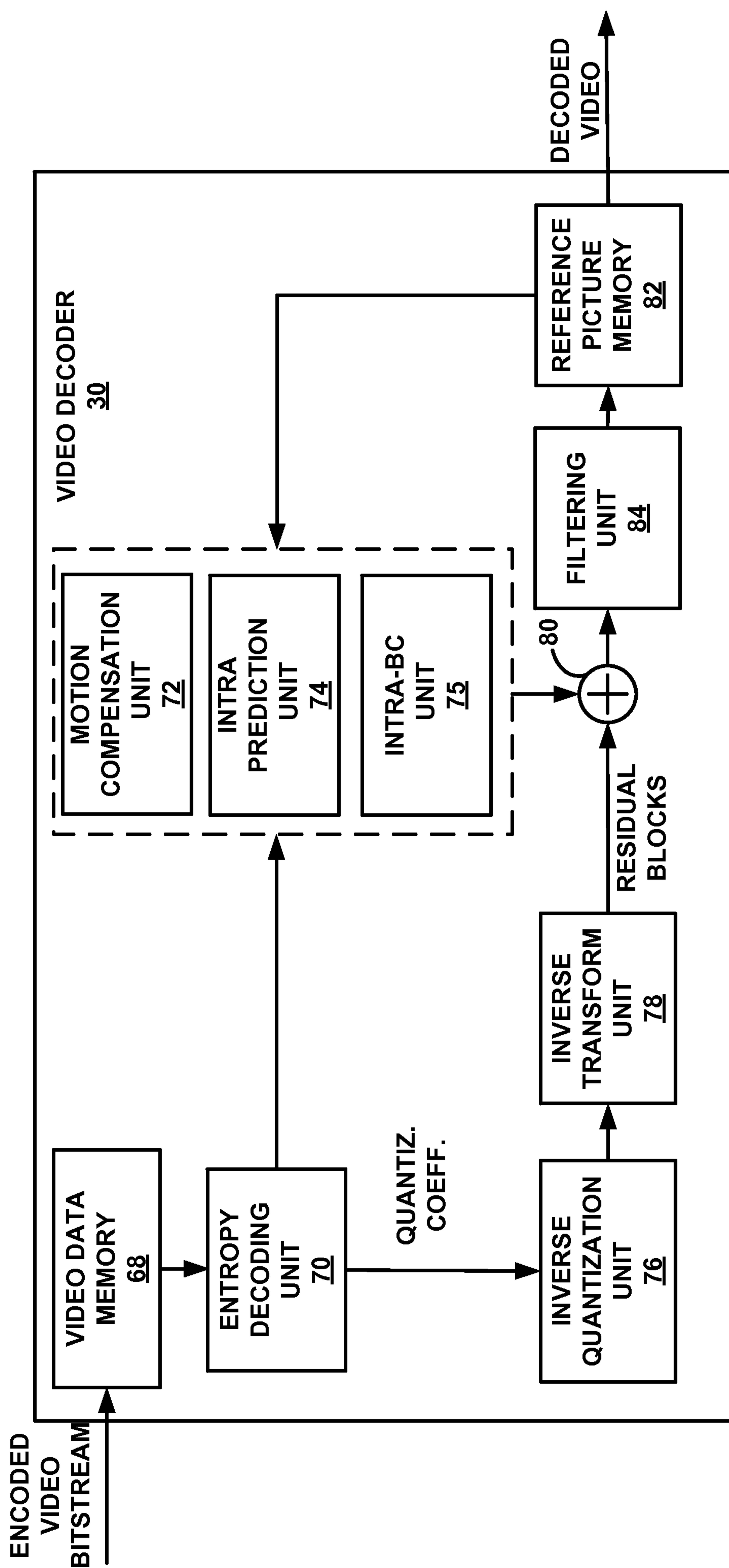


FIG. 3

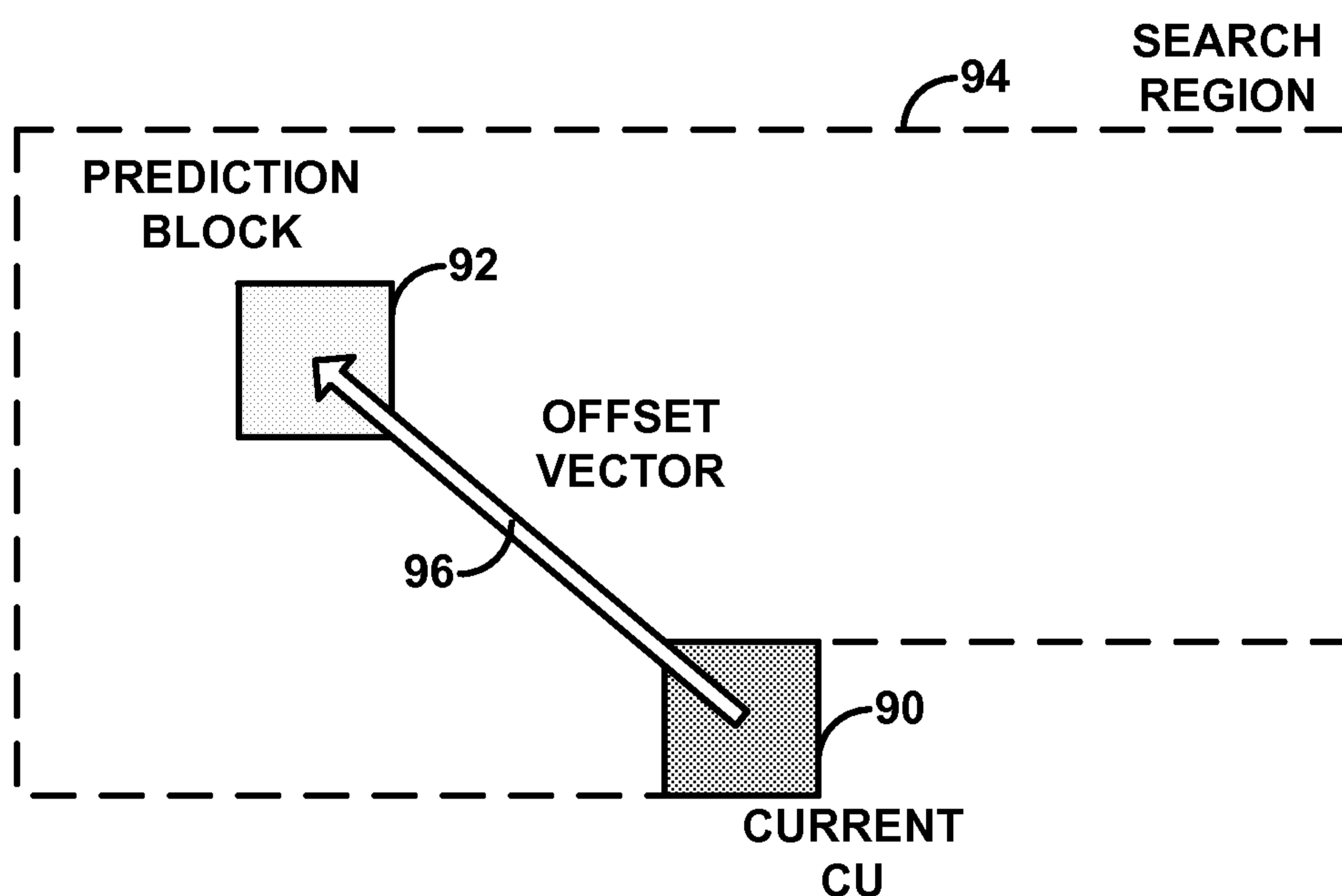


FIG. 4

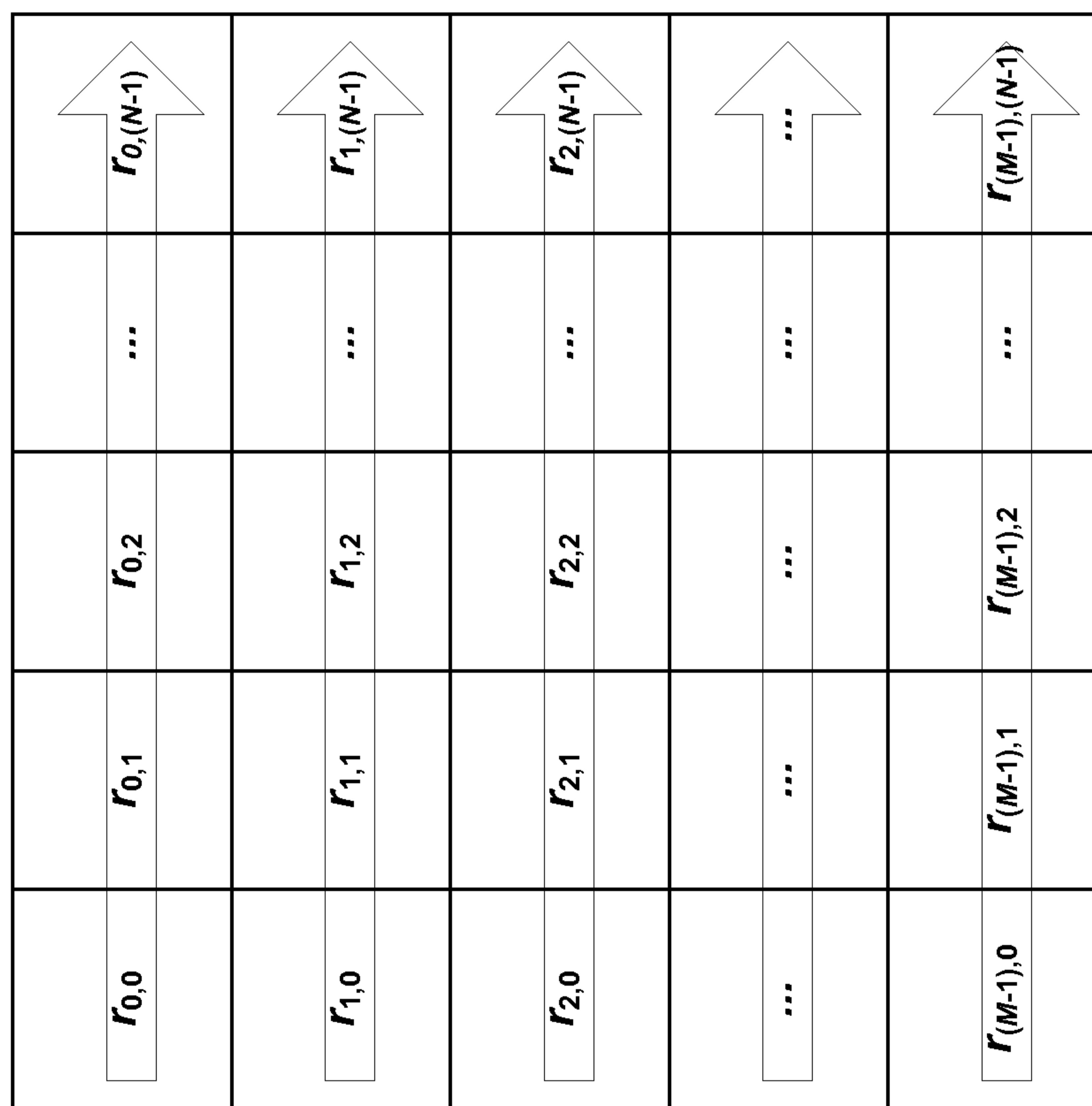


FIG. 5B

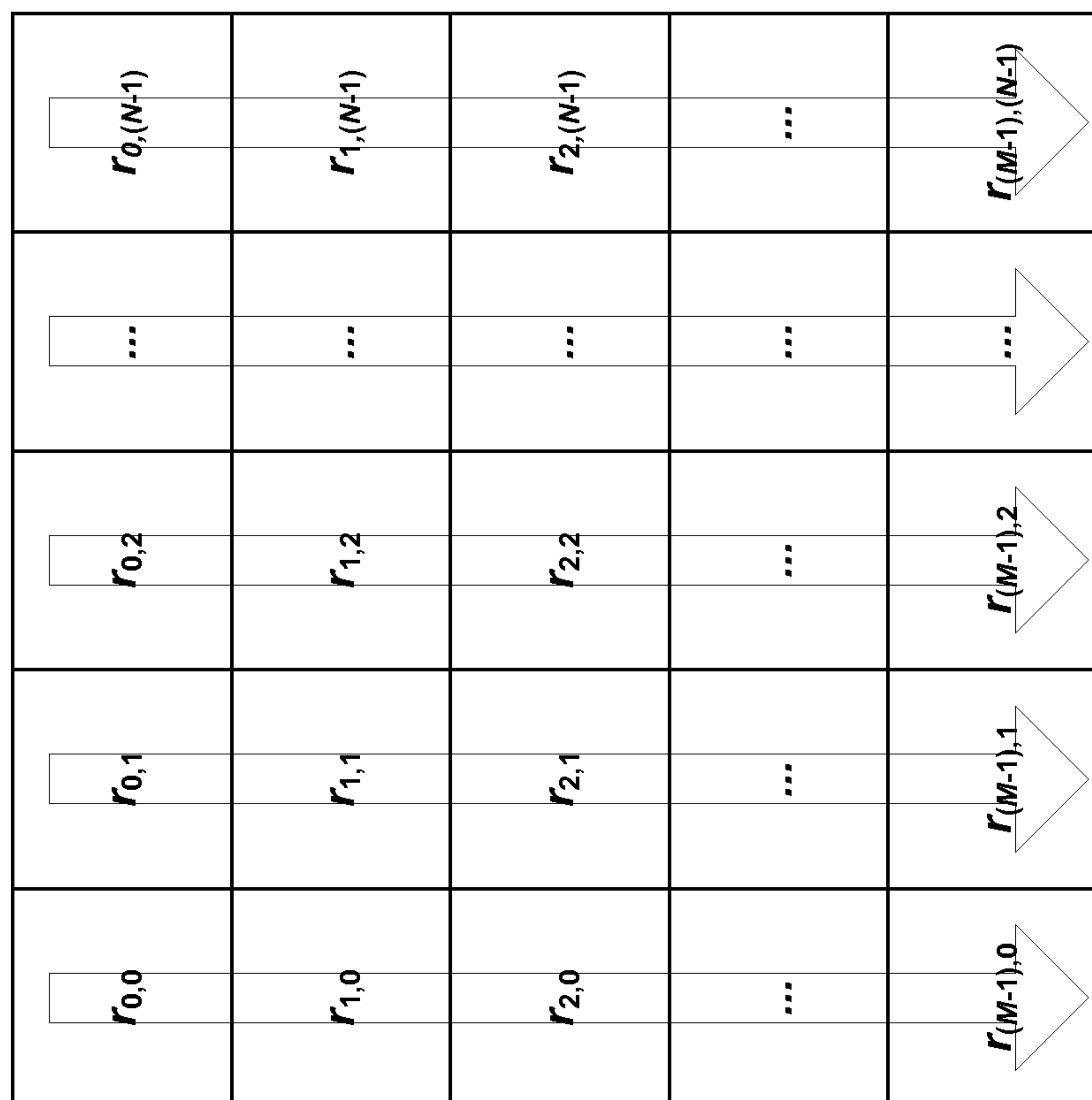


FIG. 5A

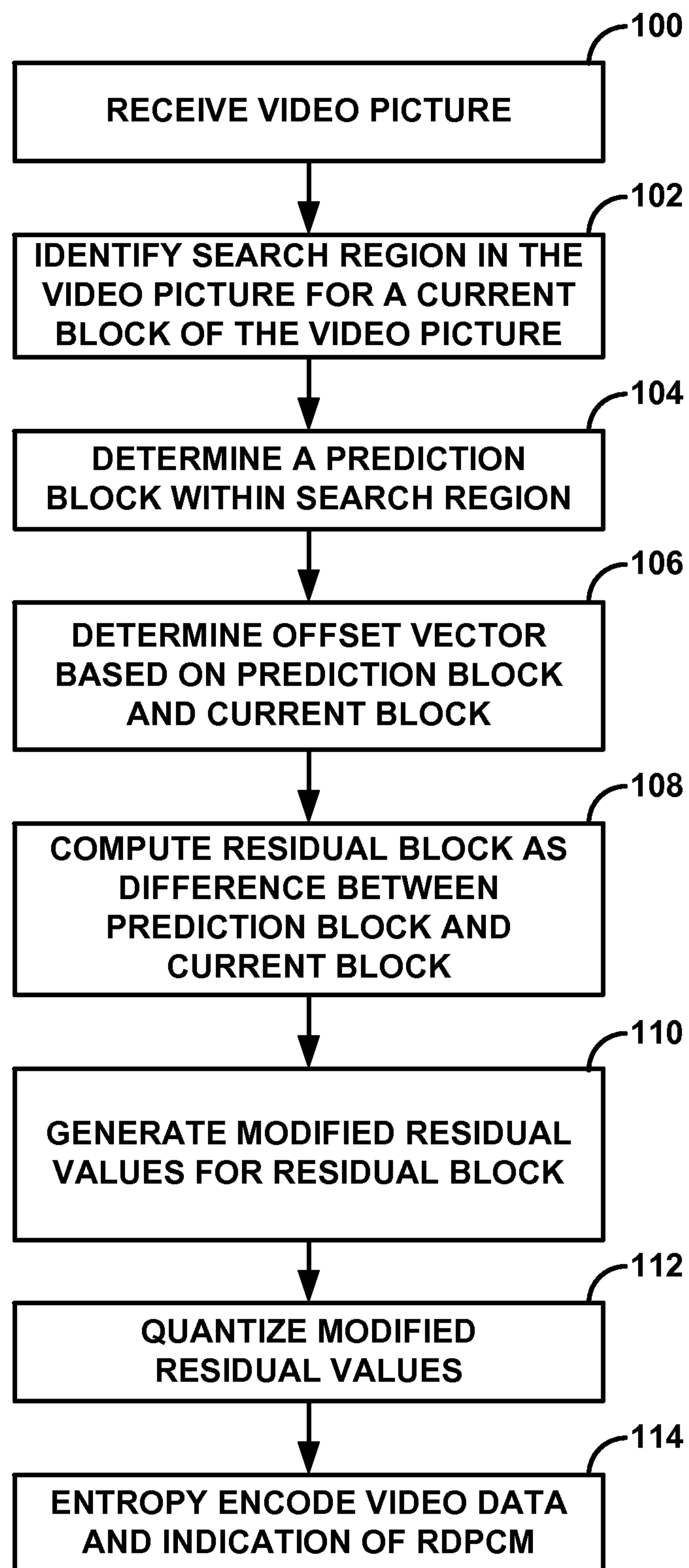


FIG. 6

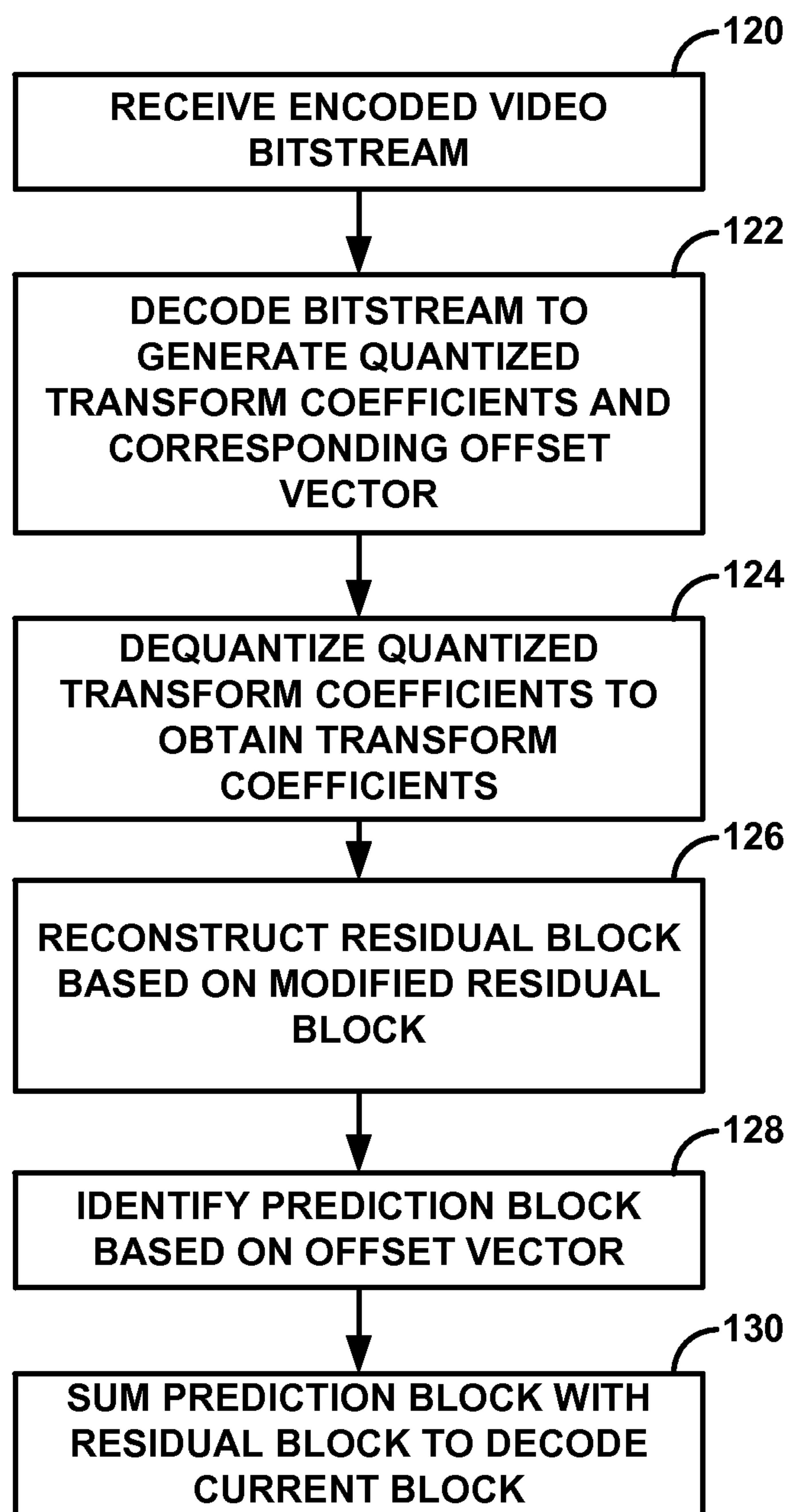


FIG. 7

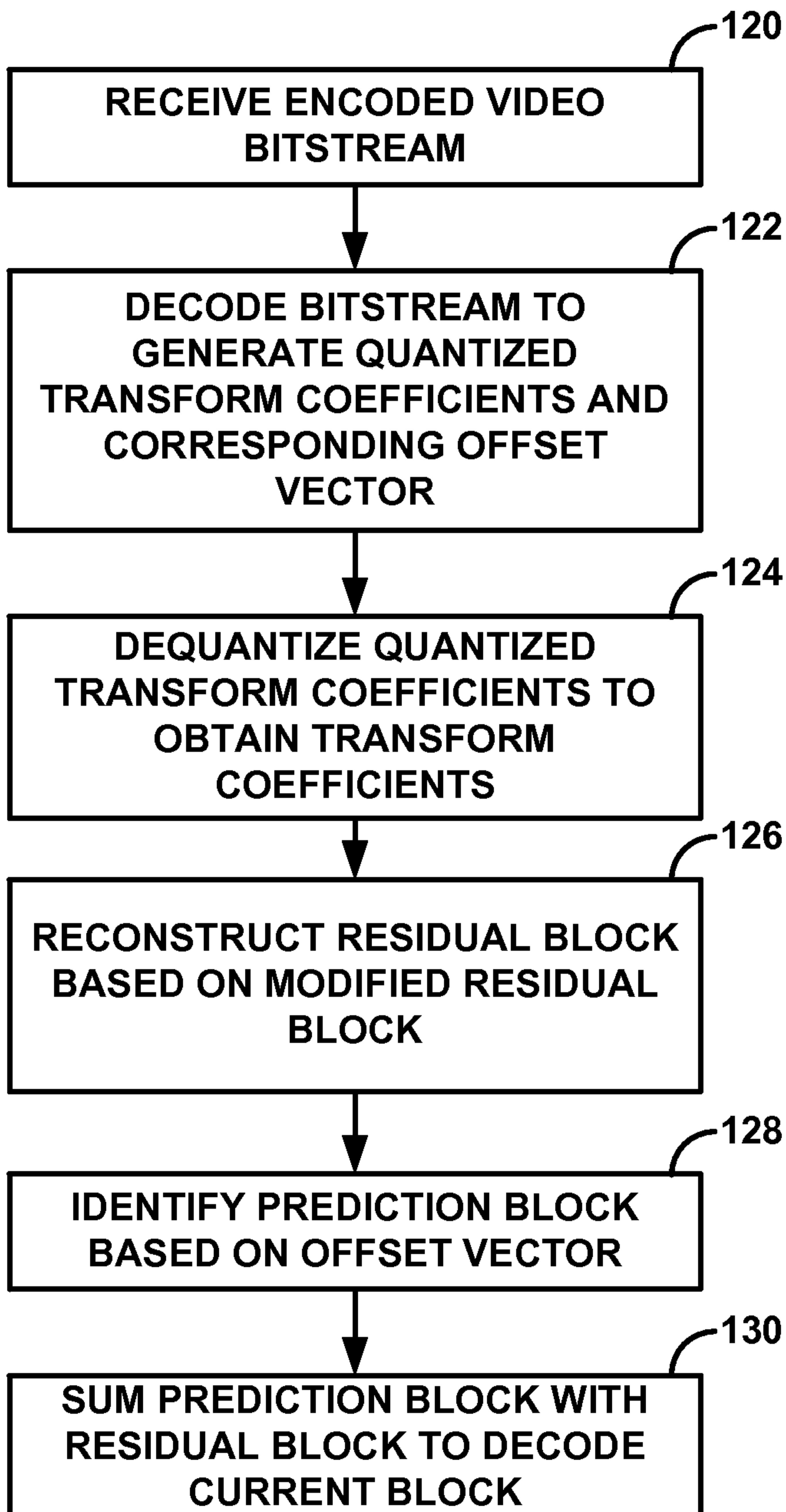


FIG. 7