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

(19) World Intellectual Property Organization
International Bureau

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

WIPO | PCT

(10) International Publication Number
WO 2015/100726 A1

(51) International Patent Classification:
H04N 19/50 (2014.01)

(21) International Application Number:
PCT/CN2014/070072

(22) International Filing Date:
3 January 2014 (03.01.2014)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **MICROSOFT CORPORATION** [US/US]; One Microsoft Way, Redmond, Washington 98052-6399 (US).

(72) Inventors; and

(71) Applicants *(for US only)*: **ZHU, Lihua** [CN/US]; c/o Microsoft Asia Pacific R&D Headquarters 14F, Building 2, No 5, Dan Ling Street Haidian District, Beijing 100080 (CN). **SULLIVAN, Gary** [US/US]; Microsoft Corporation One Microsoft Way, Redmond, Washington 98052 (US). **XU, Jizheng** [CN/CN]; c/o Microsoft Asia Pacific R&D Headquarters 14F, Building 2, No 5, Dan Ling Street Haidian District, Beijing 100080 (CN). **SANKURATRI, Sridhar** [US/US]; Microsoft Corporation One Microsoft Way, Redmond, Washington 98052 (US). **KUMAR, B. Anil** [IN/US]; Microsoft Corporation One Microsoft Way, Redmond, Washington 98052 (US). **WU, Feng** [CN/CN]; c/o Microsoft Asia Pacific R&D Headquarters 14F, Building 2, No 5, Dan Ling Street Haidian District, Beijing 100080 (CN).

[Continued on next page]

(54) Title: BLOCK VECTOR PREDICTION IN VIDEO AND IMAGE CODING/DECODING

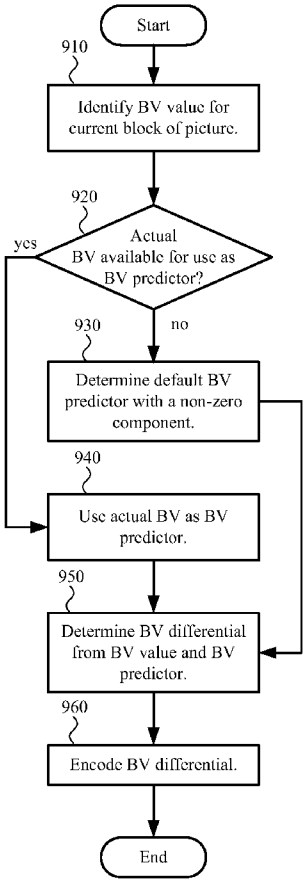


Figure 9

(57) Abstract: Innovations in the area of prediction of block vector ("BV") values improve encoding or decoding of blocks using intra block copy ("BC") prediction. For example, some of the innovations relate to use of a default BV predictor with a non-zero value. Other innovations relate to use of a selected one of multiple BV predictor candidates for a current block. Still other innovations relate to use of a skip mode in which a current intra-BC-predicted block uses a predicted BV value.

WO 2015/100726 A1

(74) **Agent**: **SHANGHAI PATENT & TRADEMARK LAW OFFICE, LLC**; 435 Guiping Road, Shanghai 200233 (CN).

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

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

Published:

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

BLOCK VECTOR PREDICTION IN VIDEO AND IMAGE CODING/DECODING

BACKGROUND

[001] Engineers use compression (also called source coding or source encoding) to reduce the bit rate of digital video. Compression decreases the cost of storing and transmitting video information by converting the information into a lower bit rate form. Decompression (also called decoding) reconstructs a version of the original information from the compressed form. A "codec" is an encoder/decoder system.

[002] Over the last two decades, various video codec standards have been adopted, including the ITU-T H.261, H.262 (MPEG-2 or ISO/IEC 13818-2), H.263 and H.264 (MPEG-4 AVC or ISO/IEC 14496-10) standards, the MPEG-1 (ISO/IEC 11172-2) and MPEG-4 Visual (ISO/IEC 14496-2) standards, and the SMPTE 421M (VC-1) standard. More recently, the HEVC standard (ITU-T H.265 or ISO/IEC 23008-2) has been approved. Extensions to the HEVC standard (*e.g.*, for scalable video coding/decoding, for coding/decoding of video with higher fidelity in terms of sample bit depth or chroma sampling rate, or for multi-view coding/decoding) are currently under development. A video codec standard typically defines options for the syntax of an encoded video bitstream, detailing parameters in the bitstream when particular features are used in encoding and decoding. In many cases, a video codec standard also provides details about the decoding operations a decoder should perform to achieve conforming results in decoding. Aside from codec standards, various proprietary codec formats define other options for the syntax of an encoded video bitstream and corresponding decoding operations.

[003] Intra block copy ("BC") is a prediction mode under development for HEVC extensions. For intra BC prediction mode, the sample values of a current block of a picture are predicted using previously reconstructed sample values in the same picture. A block vector ("BV") indicates a displacement from the current block to a region of the picture that includes the previously reconstructed sample values used for prediction. The BV is signaled in the bitstream. Intra BC prediction is a form of intra-picture prediction – intra BC prediction for a block of a picture does not use any sample values other than sample values in the same picture.

[004] As currently specified in the HEVC standard and implemented in some reference software for the HEVC standard, intra BC prediction mode has several problems. In particular, BV values are not signaled efficiently in many situations, and coding of blocks with predictable BC displacement is not efficiently handled.

SUMMARY

[005] In summary, the detailed description presents innovations in the area of prediction of block vector ("BV") values during encoding or decoding of blocks using intra block copy ("BC") prediction. For example, some of the innovations relate to use of a default BV predictor with a non-zero value. Other innovations relate to use of a selected one of multiple BV predictor candidates for a current block. Still other innovations relate to use of a skip mode in which a current intra-BC-predicted block uses a predicted BV value. In many situations, the innovations improve the coding efficiency of BV values, or otherwise improve coding efficiency for intra-BC-predicted blocks.

[006] According to a first aspect of the innovations described herein, an encoder determines a default BV predictor for a current block of a picture. The default BV predictor includes a BV predictor component with a non-zero value. For example, one of the vertical and horizontal BV components of the default BV predictor is zero, and the other has a non-zero value. The encoder encodes the current block using the default BV predictor. For example, the encoder performs intra block copy ("BC") prediction using a BV value for the current block, determines a BV differential using the BV value and the default BV predictor for the current block, then encodes the BV differential. Or, the encoder simply performs intra BC prediction using the default BV predictor.

[007] A corresponding decoder determines a default BV predictor for a current block of a picture. The default BV predictor includes a BV predictor component with a non-zero value. For example, one of the vertical and horizontal BV components of the default BV predictor is zero, and the other has a non-zero value. The decoder decodes the current block using the default BV predictor. For example, the decoder decodes a BV differential for the current block, then combines the decoded BV differential with the default BV predictor to

reconstruct a BV value for the current block. The decoder performs intra BC prediction using the BV value. Or, the decoder simply performs intra BC prediction using the default BV predictor.

[008] The encoder or decoder may conditionally use the default BV predictor depending on whether an actual BV value for a previous block of the picture is available. For example, the encoder/decoder checks whether an actual BV value for a previous block of the picture is available. If so, the actual BV value for the previous block is used as the BV predictor for the current block. Otherwise, the default BV predictor is used as the BV predictor for the current block. In some implementations, an actual BV value for a previous block is available if (1) the previous block and the current block are part of a given coding tree unit, and (2) prediction mode of the previous block is intra BC prediction mode.

[009] According to a second aspect of the innovations described herein, an encoder uses BV prediction to determine a BV value for a current block of a picture. The BV value for the current block indicates a displacement to a region within the picture. The encoder encodes the current block using intra BC with the BV value (here, the BV predictor). The encoder outputs in a bitstream an indication that the current block is encoded in skip mode, and the bitstream lacks a BV differential and residual data for the current block.

[010] A corresponding decoder receives from a bitstream an indication that a current block of a picture is encoded in skip mode. The bitstream lacks a BV differential and residual data for the current block. The decoder uses BV prediction to determine a BV value for the current block. The decoder then decodes the current block using intra block copy prediction with the BV value (here, the BV predictor).

[011] For the skipped block, the bitstream can include an index value that indicates a selection of a BV predictor candidate, from a set of multiple BV predictor candidates, to use as the BV value (here, the BV predictor) for the current block. The index value can be a flag value or integer value.

[012] According to a third aspect of the innovations described herein, an encoder determines a set of multiple BV predictor candidates for a current block of a picture, then selects one of the BV predictor candidates to use for the current block. The encoder encodes

the current block using the selected BV predictor candidate, outputting in a bitstream an index value that indicates the selected BV predictor candidate.

[013] A corresponding decoder receives from a bitstream an index value. The decoder determines a set of multiple BV predictor candidates for a current block of a picture and, based on the index value, selects one of the BV predictor candidates to use for the current block. The decoder decodes the current block using the selected BV predictor candidate.

[014] In some cases, the bitstream includes a BV differential for the current block (which is a non-skip mode, non-merge-mode block). The BV differential indicates a difference between the selected BV predictor candidate and the BV value for the current block. In other cases, the current block is a merge-mode block, and the bitstream lacks a BV differential for the current block, but may include residual data for the current block. In still other cases, a value in the bitstream indicates the current block is a skip-mode block. In these cases, the bitstream lacks a BV differential for the current block, and the bitstream lacks residual data for the current block.

[015] In general, the multiple BV predictor candidates include up to x actual BV values of previous blocks. The previous blocks can be identified based at least in part on location in a neighborhood around the current block (*e.g.*, blocks to the left of, above and to the above-right of the current block). Or, the previous blocks can be identified based at least in part on decoding order compared to the current block (*e.g.*, last three intra-BC-predicted blocks). The multiple BV predictor candidates can also include one or more default BV predictors.

[016] The innovations for BV prediction can be implemented as part of a method, as part of a computing device adapted to perform the method or as part of a tangible computer-readable media storing computer-executable instructions for causing a computing device to perform the method. The various innovations can be used in combination or separately. In particular, merge mode for BV prediction can be used in conjunction with default BV predictors with non-zero motion and/or skip mode for intra-BC-predicted blocks. Skip mode for intra-BC-predicted blocks can be used in conjunction with default BV predictors with non-zero motion.

[017] The foregoing and other objects, features, and advantages of the invention will become more apparent from the following detailed description, which proceeds with reference to the accompanying figures.

BRIEF DESCRIPTION OF THE DRAWINGS

[018] Figure 1 is a diagram of an example computing system in which some described embodiments can be implemented.

[019] Figures 2a and 2b are diagrams of example network environments in which some described embodiments can be implemented.

[020] Figure 3 is a diagram of an example encoder system in conjunction with which some described embodiments can be implemented.

[021] Figure 4 is a diagram of an example decoder system in conjunction with which some described embodiments can be implemented.

[022] Figures 5a and 5b are diagrams illustrating an example video encoder in conjunction with which some described embodiments can be implemented.

[023] Figure 6 is a diagram illustrating an example video decoder in conjunction with which some described embodiments can be implemented.

[024] Figure 7a – 7d are diagrams illustrating intra BC prediction for a block of a picture, BV prediction for the block, a default (0, 0) BV predictor, and a default BV predictor with a non-zero value, respectively.

[025] Figures 8 and 9 are flowcharts illustrating techniques for encoding that includes use of a default BV predictor with a non-zero value.

[026] Figures 10 and 11 are flowcharts illustrating techniques for decoding that includes use of a default BV predictor with a non-zero value.

[027] Figures 12 and 13 are flowcharts illustrating techniques for encoding and decoding, respectively, that includes skip mode for intra-BC-predicted blocks.

[028] Figure 14 is a diagram illustrating BV predictor candidates among actual BV values of previous blocks in a picture.

[029] Figure 15 is a diagram illustrating BV predictor candidates among actual BV values of neighboring blocks in a picture.

[030] Figures 16a and 16b are flowcharts illustrating techniques for encoding that includes selection among multiple BV predictor candidates.

[031] Figures 17a and 17b are flowcharts illustrating techniques for decoding that includes selection among multiple BV predictor candidates.

DETAILED DESCRIPTION

[032] The detailed description presents innovations in the area of prediction of block vector ("BV") values during encoding or decoding of blocks using intra block copy ("BC") prediction. For example, some of the innovations relate to use of a default BV predictor with a non-zero value. Other innovations relate to use of a selected one of multiple BV predictor candidates for a current block. Still other innovations relate to use of a skip mode in which a current intra-BC-predicted block uses a predicted BV value. In many situations, the innovations improve the coding efficiency of BV values, which can lower bit rate and/or improve quality, or otherwise improve coding efficiency for intra-BC-predicted blocks.

[033] Although operations described herein are in places described as being performed by a video encoder or decoder, in many cases the operations can be performed by another type of media processing tool (*e.g.*, image encoder or decoder).

[034] Some of the innovations described herein are illustrated with reference to syntax elements and operations specific to the HEVC standard. For example, reference is made to the draft version JCTVC-O1005 of the HEVC standard – "High Efficiency Video Coding

(HEVC) Range Extensions Text Specification: Draft 5," JCTVC-O1005_v3, November 2013. The innovations described herein can also be implemented for other standards or formats.

[035] More generally, various alternatives to the examples described herein are possible. For example, some of the methods described herein can be altered by changing the ordering of the method acts described, by splitting, repeating, or omitting certain method acts, etc. The various aspects of the disclosed technology can be used in combination or separately. Different embodiments use one or more of the described innovations. Some of the innovations described herein address one or more of the problems noted in the background. Typically, a given technique/tool does not solve all such problems.

I. Example Computing Systems.

[036] Figure 1 illustrates a generalized example of a suitable computing system (100) in which several of the described innovations may be implemented. The computing system (100) is not intended to suggest any limitation as to scope of use or functionality, as the innovations may be implemented in diverse general-purpose or special-purpose computing systems.

[037] With reference to Figure 1, the computing system (100) includes one or more processing units (110, 115) and memory (120, 125). The processing units (110, 115) execute computer-executable instructions. A processing unit can be a general-purpose central processing unit ("CPU"), processor in an application-specific integrated circuit ("ASIC") or any other type of processor. In a multi-processing system, multiple processing units execute computer-executable instructions to increase processing power. For example, Figure 1 shows a central processing unit (110) as well as a graphics processing unit or co-processing unit (115). The tangible memory (120, 125) may be volatile memory (*e.g.*, registers, cache, RAM), non-volatile memory (*e.g.*, ROM, EEPROM, flash memory, etc.), or some combination of the two, accessible by the processing unit(s). The memory (120, 125) stores software (180) implementing one or more innovations for block vector prediction, in the form of computer-executable instructions suitable for execution by the processing unit(s).

[038] A computing system may have additional features. For example, the computing system (100) includes storage (140), one or more input devices (150), one or more output devices (160), and one or more communication connections (170). An interconnection mechanism (not shown) such as a bus, controller, or network interconnects the components of the computing system (100). Typically, operating system software (not shown) provides an operating environment for other software executing in the computing system (100), and coordinates activities of the components of the computing system (100).

[039] The tangible storage (140) may be removable or non-removable, and includes magnetic disks, magnetic tapes or cassettes, CD-ROMs, DVDs, or any other medium which can be used to store information and which can be accessed within the computing system (100). The storage (140) stores instructions for the software (180) implementing one or more innovations for block vector prediction.

[040] The input device(s) (150) may be a touch input device such as a keyboard, mouse, pen, or trackball, a voice input device, a scanning device, or another device that provides input to the computing system (100). For video, the input device(s) (150) may be a camera, video card, TV tuner card, or similar device that accepts video input in analog or digital form, or a CD-ROM or CD-RW that reads video input into the computing system (100). The output device(s) (160) may be a display, printer, speaker, CD-writer, or another device that provides output from the computing system (100).

[041] The communication connection(s) (170) enable communication over a communication medium to another computing entity. The communication medium conveys information such as computer-executable instructions, audio or video input or output, or other data in a modulated data signal. A modulated data signal is a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media can use an electrical, optical, RF, or other carrier.

[042] The innovations can be described in the general context of computer-readable media. Computer-readable media are any available tangible media that can be accessed within a computing environment. By way of example, and not limitation, with the computing

system (100), computer-readable media include memory (120, 125), storage (140), and combinations of any of the above.

[043] The innovations can be described in the general context of computer-executable instructions, such as those included in program modules, being executed in a computing system on a target real or virtual processor. Generally, program modules include routines, programs, libraries, objects, classes, components, data structures, etc. that perform particular tasks or implement particular abstract data types. The functionality of the program modules may be combined or split between program modules as desired in various embodiments. Computer-executable instructions for program modules may be executed within a local or distributed computing system.

[044] The terms "system" and "device" are used interchangeably herein. Unless the context clearly indicates otherwise, neither term implies any limitation on a type of computing system or computing device. In general, a computing system or computing device can be local or distributed, and can include any combination of special-purpose hardware and/or general-purpose hardware with software implementing the functionality described herein.

[045] The disclosed methods can also be implemented using specialized computing hardware configured to perform any of the disclosed methods. For example, the disclosed methods can be implemented by an integrated circuit (*e.g.*, an ASIC (such as an ASIC digital signal processor ("DSP"), a graphics processing unit ("GPU"), or a programmable logic device ("PLD"), such as a field programmable gate array ("FPGA")) specially designed or configured to implement any of the disclosed methods.

[046] For the sake of presentation, the detailed description uses terms like "determine" and "use" to describe computer operations in a computing system. These terms are high-level abstractions for operations performed by a computer, and should not be confused with acts performed by a human being. The actual computer operations corresponding to these terms vary depending on implementation.

II. Example Network Environments.

[047] Figures 2a and 2b show example network environments (201, 202) that include video encoders (220) and video decoders (270). The encoders (220) and decoders (270) are connected over a network (250) using an appropriate communication protocol. The network (250) can include the Internet or another computer network.

[048] In the network environment (201) shown in Figure 2a, each real-time communication ("RTC") tool (210) includes both an encoder (220) and a decoder (270) for bidirectional communication. A given encoder (220) can produce output compliant with a variation or extension of the HEVC standard (also known as H.265), SMPTE 421M standard, ISO-IEC 14496-10 standard (also known as H.264 or AVC), another standard, or a proprietary format, with a corresponding decoder (270) accepting encoded data from the encoder (220). The bidirectional communication can be part of a video conference, video telephone call, or other two-party or multi-part communication scenario. Although the network environment (201) in Figure 2a includes two real-time communication tools (210), the network environment (201) can instead include three or more real-time communication tools (210) that participate in multi-party communication.

[049] A real-time communication tool (210) manages encoding by an encoder (220). Figure 3 shows an example encoder system (300) that can be included in the real-time communication tool (210). Alternatively, the real-time communication tool (210) uses another encoder system. A real-time communication tool (210) also manages decoding by a decoder (270). Figure 4 shows an example decoder system (400), which can be included in the real-time communication tool (210). Alternatively, the real-time communication tool (210) uses another decoder system.

[050] In the network environment (202) shown in Figure 2b, an encoding tool (212) includes an encoder (220) that encodes video for delivery to multiple playback tools (214), which include decoders (270). The unidirectional communication can be provided for a video surveillance system, web camera monitoring system, remote desktop conferencing presentation or other scenario in which video is encoded and sent from one location to one or more other locations. Although the network environment (202) in Figure 2b includes two

playback tools (214), the network environment (202) can include more or fewer playback tools (214). In general, a playback tool (214) communicates with the encoding tool (212) to determine a stream of video for the playback tool (214) to receive. The playback tool (214) receives the stream, buffers the received encoded data for an appropriate period, and begins decoding and playback.

[051] Figure 3 shows an example encoder system (300) that can be included in the encoding tool (212). Alternatively, the encoding tool (212) uses another encoder system. The encoding tool (212) can also include server-side controller logic for managing connections with one or more playback tools (214). Figure 4 shows an example decoder system (400), which can be included in the playback tool (214). Alternatively, the playback tool (214) uses another decoder system. A playback tool (214) can also include client-side controller logic for managing connections with the encoding tool (212).

III. Example Encoder Systems.

[052] Figure 3 is a block diagram of an example encoder system (300) in conjunction with which some described embodiments may be implemented. The encoder system (300) can be a general-purpose encoding tool capable of operating in any of multiple encoding modes such as a low-latency encoding mode for real-time communication, a transcoding mode, and a higher-latency encoding mode for producing media for playback from a file or stream, or it can be a special-purpose encoding tool adapted for one such encoding mode. The encoder system (300) can be implemented as an operating system module, as part of an application library or as a standalone application. Overall, the encoder system (300) receives a sequence of source video frames (311) from a video source (310) and produces encoded data as output to a channel (390). The encoded data output to the channel can include content encoded using innovations in BV prediction as described herein.

[053] The video source (310) can be a camera, tuner card, storage media, or other digital video source. The video source (310) produces a sequence of video frames at a frame rate of, for example, 30 frames per second. As used herein, the term “frame” generally refers to source, coded or reconstructed image data. For progressive-scan video, a frame is a

progressive-scan video frame. For interlaced video, in example embodiments, an interlaced video frame is de-interlaced prior to encoding. Alternatively, two complementary interlaced video fields are encoded together as a single video frame or encoded as two separately-encoded fields. Aside from indicating a progressive-scan video frame or interlaced-scan video frame, the term "frame" or "picture" can indicate a single non-paired video field, a complementary pair of video fields, a video object plane that represents a video object at a given time, or a region of interest in a larger image. The video object plane or region can be part of a larger image that includes multiple objects or regions of a scene.

[054] An arriving source frame (311) is stored in a source frame temporary memory storage area (320) that includes multiple frame buffer storage areas (321, 322, … , 32*n*). A frame buffer (321, 322, etc.) holds one source frame in the source frame storage area (320). After one or more of the source frames (311) have been stored in frame buffers (321, 322, etc.), a frame selector (330) selects an individual source frame from the source frame storage area (320). The order in which frames are selected by the frame selector (330) for input to the encoder (340) may differ from the order in which the frames are produced by the video source (310), *e.g.*, the encoding of some frames may be delayed in order, so as to allow some later frames to be encoded first and to thus facilitate temporally backward prediction. Before the encoder (340), the encoder system (300) can include a pre-processor (not shown) that performs pre-processing (*e.g.*, filtering) of the selected frame (331) before encoding. The pre-processing can include color space conversion into primary (*e.g.*, luma) and secondary (*e.g.*, chroma differences toward red and toward blue) components and resampling processing (*e.g.*, to reduce the spatial resolution of chroma components) for encoding. Typically, before encoding, video has been converted to a color space such as YUV, in which sample values of a luma (Y) component represent brightness or intensity values, and sample values of chroma (U, V) components represent color-difference values. The chroma sample values may be sub-sampled to a lower chroma sampling rate (*e.g.*, for YUV 4:2:0 format), or the chroma sample values may have the same resolution as the luma sample values (*e.g.*, for YUV 4:4:4 format). Or, the video can be encoded in another format (*e.g.*, RGB 4:4:4 format).

[055] The encoder (340) encodes the selected frame (331) to produce a coded frame (341) and also produces memory management control operation ("MMCO") signals (342) or

reference picture set ("RPS") information. If the current frame is not the first frame that has been encoded, when performing its encoding process, the encoder (340) may use one or more previously encoded/decoded frames (369) that have been stored in a decoded frame temporary memory storage area (360). Such stored decoded frames (369) are used as reference frames for inter-frame prediction of the content of the current source frame (331). The MMCO/RPS information (342) indicates to a decoder which reconstructed frames may be used as reference frames, and hence should be stored in a frame storage area.

[056] Generally, the encoder (340) includes multiple encoding modules that perform encoding tasks such as partitioning into tiles, intra prediction estimation and prediction, motion estimation and compensation, frequency transforms, quantization and entropy coding. The exact operations performed by the encoder (340) can vary depending on compression format. The format of the output encoded data can be a variation or extension of HEVC format (H.265), Windows Media Video format, VC-1 format, MPEG-x format (*e.g.*, MPEG-1, MPEG-2, or MPEG-4), H.26x format (*e.g.*, H.261, H.262, H.263, H.264), or another format.

[057] The encoder (340) can partition a frame into multiple tiles of the same size or different sizes. For example, the encoder (340) splits the frame along tile rows and tile columns that, with frame boundaries, define horizontal and vertical boundaries of tiles within the frame, where each tile is a rectangular region. Tiles are often used to provide options for parallel processing. A frame can also be organized as one or more slices, where a slice can be an entire frame or region of the frame. A slice can be decoded independently of other slices in a frame, which improves error resilience. The content of a slice or tile is further partitioned into blocks or other sets of samples for purposes of encoding and decoding.

[058] For syntax according to the HEVC standard, the encoder splits the content of a frame (or slice or tile) into coding tree units. A coding tree unit ("CTU") includes luma sample values organized as a luma coding tree block ("CTB") and corresponding chroma sample values organized as two chroma CTBs. The size of a CTU (and its CTBs) is selected by the encoder, and can be, for example, 64x64, 32x32 or 16x16 sample values. A CTU includes one or more coding units. A coding unit ("CU") has a luma coding block ("CB") and two corresponding chroma CBs. For example, a CTU with a 64x64 luma CTB and two 64x64 chroma CTBs (YUV 4:4:4 format) can be split into four CUs, with each CU including

a 32x32 luma CB and two 32x32 chroma CBs, and with each CU possibly being split further into smaller CUs. Or, as another example, a CTU with a 64x64 luma CTB and two 32x32 chroma CTBs (YUV 4:2:0 format) can be split into four CUs, with each CU including a 32x32 luma CB and two 16x16 chroma CBs, and with each CU possibly being split further into smaller CUs. The smallest allowable size of CU (*e.g.*, 8x8, 16x16) can be signaled in the bitstream.

[059] Generally, a CU has a prediction mode such as inter or intra. A CU includes one or more prediction units for purposes of signaling of prediction information (such as prediction mode details, displacement values, etc.) and/or prediction processing. A prediction unit ("PU") has a luma prediction block ("PB") and two chroma PBs. For an intra-predicted CU, the PU has the same size as the CU, unless the CU has the smallest size (*e.g.*, 8x8). In that case, the CU can be split into four smaller PUs (*e.g.*, each 4x4 if the smallest CU size is 8x8) or the PU can have the smallest CU size, as indicated by a syntax element for the CU. A CU also has one or more transform units for purposes of residual coding/decoding, where a transform unit ("TU") has a transform block ("TB") and two chroma TBs. A PU in an intra-predicted CU may contain a single TU (equal in size to the PU) or multiple TUs. As used herein, the term "block" can indicate a CB, PB, TB or other set of sample values, depending on context. The encoder decides how to partition video into CTUs, CUs, PUs, TUs, etc.

[060] Returning to Figure 3, the encoder represents an intra-coded block of a source frame (331) in terms of prediction from other, previously reconstructed sample values in the frame (331). For intra BC prediction, an intra-picture estimator estimates displacement of a block with respect to the other, previously reconstructed sample values. An intra-frame prediction reference region is a region of samples in the frame that are used to generate BC-prediction values for the block. The intra-frame prediction region can be indicated with a block vector ("BV") value (determined in BV estimation). For intra spatial prediction for a block, the intra-picture estimator estimates extrapolation of the neighboring reconstructed sample values into the block. The intra-picture estimator can output prediction information (such as BV values for intra BC prediction or prediction mode (direction) for intra spatial

prediction), which is entropy coded. An intra-frame prediction predictor applies the prediction information to determine intra prediction values.

[061] The encoder (340) represents an inter-frame coded, predicted block of a source frame (331) in terms of prediction from reference frames. A motion estimator estimates the motion of the block with respect to one or more reference frames (369). When multiple reference frames are used, the multiple reference frames can be from different temporal directions or the same temporal direction. A motion-compensated prediction reference region is a region of samples in the reference frame(s) that are used to generate motion-compensated prediction values for a block of samples of a current frame. The motion estimator outputs motion information such as motion vector ("MV") information, which is entropy coded. A motion compensator applies MVs to reference frames (369) to determine motion-compensated prediction values for inter-frame prediction.

[062] The encoder can determine the differences (if any) between a block's prediction values (intra or inter) and corresponding original values. These prediction residual values are further encoded using a frequency transform, quantization and entropy encoding. For example, the encoder (340) sets values for quantization parameter ("QP") for a picture, tile, slice and/or other portion of video, and quantizes transform coefficients accordingly. The entropy coder of the encoder (340) compresses quantized transform coefficient values as well as certain side information (*e.g.*, MV information, index values for BV predictors, BV differentials, QP values, mode decisions, parameter choices). Typical entropy coding techniques include Exponential-Golomb coding, Golomb-Rice coding, arithmetic coding, differential coding, Huffman coding, run length coding, variable-length-to-variable-length ("V2V") coding, variable-length-to-fixed-length ("V2F") coding, Lempel-Ziv ("LZ") coding, dictionary coding, probability interval partitioning entropy coding ("PIPE"), and combinations of the above. The entropy coder can use different coding techniques for different kinds of information, can apply multiple techniques in combination (*e.g.*, by applying Golomb-Rice coding followed by arithmetic coding), and can choose from among multiple code tables within a particular coding technique.

[063] An adaptive deblocking filter is included within the motion compensation loop in the encoder (340) to smooth discontinuities across block boundary rows and/or columns in a

decoded frame. Other filtering (such as de-ringing filtering, adaptive loop filtering ("ALF"), or sample-adaptive offset ("SAO") filtering; not shown) can alternatively or additionally be applied as in-loop filtering operations.

[064] The coded frames (341) and MMCO/RPS information (342) (or information equivalent to the MMCO/RPS information (342), since the dependencies and ordering structures for frames are already known at the encoder (340)) are processed by a decoding process emulator (350). The decoding process emulator (350) implements some of the functionality of a decoder, for example, decoding tasks to reconstruct reference frames. In a manner consistent with the MMCO/RPS information (342), the decoding processes emulator (350) determines whether a given coded frame (341) needs to be reconstructed and stored for use as a reference frame in inter-frame prediction of subsequent frames to be encoded. If a coded frame (341) needs to be stored, the decoding process emulator (350) models the decoding process that would be conducted by a decoder that receives the coded frame (341) and produces a corresponding decoded frame (351). In doing so, when the encoder (340) has used decoded frame(s) (369) that have been stored in the decoded frame storage area (360), the decoding process emulator (350) also uses the decoded frame(s) (369) from the storage area (360) as part of the decoding process.

[065] The decoded frame temporary memory storage area (360) includes multiple frame buffer storage areas (361, 362, ..., 36*n*). In a manner consistent with the MMCO/RPS information (342), the decoding process emulator (350) manages the contents of the storage area (360) in order to identify any frame buffers (361, 362, etc.) with frames that are no longer needed by the encoder (340) for use as reference frames. After modeling the decoding process, the decoding process emulator (350) stores a newly decoded frame (351) in a frame buffer (361, 362, etc.) that has been identified in this manner.

[066] The coded frames (341) and MMCO/RPS information (342) are buffered in a temporary coded data area (370). The coded data that is aggregated in the coded data area (370) contains, as part of the syntax of an elementary coded video bitstream, encoded data for one or more pictures. The coded data that is aggregated in the coded data area (370) can also include media metadata relating to the coded video data (*e.g.*, as one or more parameters in

one or more supplemental enhancement information ("SEI") messages or video usability information ("VUI") messages).

[067] The aggregated data (371) from the temporary coded data area (370) are processed by a channel encoder (380). The channel encoder (380) can packetize and/or multiplex the aggregated data for transmission or storage as a media stream (*e.g.*, according to a media program stream or transport stream format such as ITU-T H.222.0 | ISO/IEC 13818-1 or an Internet real-time transport protocol format such as IETF RFC 3550), in which case the channel encoder (380) can add syntax elements as part of the syntax of the media transmission stream. Or, the channel encoder (380) can organize the aggregated data for storage as a file (*e.g.*, according to a media container format such as ISO/IEC 14496-12), in which case the channel encoder (380) can add syntax elements as part of the syntax of the media storage file. Or, more generally, the channel encoder (380) can implement one or more media system multiplexing protocols or transport protocols, in which case the channel encoder (380) can add syntax elements as part of the syntax of the protocol(s). The channel encoder (380) provides output to a channel (390), which represents storage, a communications connection, or another channel for the output. The channel encoder (380) or channel (390) may also include other elements (not shown), *e.g.*, for forward-error correction ("FEC") encoding and analog signal modulation.

IV. Example Decoder Systems.

[068] Figure 4 is a block diagram of an example decoder system (400) in conjunction with which some described embodiments may be implemented. The decoder system (400) can be a general-purpose decoding tool capable of operating in any of multiple decoding modes such as a low-latency decoding mode for real-time communication and a higher-latency decoding mode for media playback from a file or stream, or it can be a special-purpose decoding tool adapted for one such decoding mode. The decoder system (400) can be implemented as an operating system module, as part of an application library or as a standalone application. Overall, the decoder system (400) receives coded data from a channel (410) and produces reconstructed frames as output for an output destination (490).

The coded data can include content that has been encoded using innovations in BV prediction as described herein.

[069] The decoder system (400) includes a channel (410), which can represent storage, a communications connection, or another channel for coded data as input. The channel (410) produces coded data that has been channel coded. A channel decoder (420) can process the coded data. For example, the channel decoder (420) de-packetizes and/or demultiplexes data that has been aggregated for transmission or storage as a media stream (*e.g.*, according to a media program stream or transport stream format such as ITU-T H.222.0 | ISO/IEC 13818-1 or an internet real-time transport protocol format such as IETF RFC 3550), in which case the channel decoder (420) can parse syntax elements added as part of the syntax of the media transmission stream. Or, the channel decoder (420) separates coded video data that has been aggregated for storage as a file (*e.g.*, according to a media container format such as ISO/IEC 14496-12), in which case the channel decoder (420) can parse syntax elements added as part of the syntax of the media storage file. Or, more generally, the channel decoder (420) can implement one or more media system demultiplexing protocols or transport protocols, in which case the channel decoder (420) can parse syntax elements added as part of the syntax of the protocol(s). The channel (410) or channel decoder (420) may also include other elements (not shown), *e.g.*, for FEC decoding and analog signal demodulation.

[070] The coded data (421) that is output from the channel decoder (420) is stored in a temporary coded data area (430) until a sufficient quantity of such data has been received. The coded data (421) includes coded frames (431) and MMCO/RPS information (432). The coded data (421) in the coded data area (430) contain, as part of the syntax of an elementary coded video bitstream, coded data for one or more pictures. The coded data (421) in the coded data area (430) can also include media metadata relating to the encoded video data (*e.g.*, as one or more parameters in one or more SEI messages or VUI messages).

[071] In general, the coded data area (430) temporarily stores coded data (421) until such coded data (421) is used by the decoder (450). At that point, coded data for a coded frame (431) and MMCO/RPS information (432) are transferred from the coded data area (430) to the decoder (450). As decoding continues, new coded data is added to the coded data area

(430) and the oldest coded data remaining in the coded data area (430) is transferred to the decoder (450).

[072] The decoder (450) decodes a coded frame (431) to produce a corresponding decoded frame (451). As appropriate, when performing its decoding process, the decoder (450) may use one or more previously decoded frames (469) as reference frames for inter-frame prediction. The decoder (450) reads such previously decoded frames (469) from a decoded frame temporary memory storage area (460). Generally, the decoder (450) includes multiple decoding modules that perform decoding tasks such as entropy decoding, intra-frame prediction, motion-compensated inter-frame prediction, inverse quantization, inverse frequency transforms, and merging of tiles. The exact operations performed by the decoder (450) can vary depending on compression format.

[073] For example, the decoder (450) receives encoded data for a compressed frame or sequence of frames and produces output including decoded frame (451). In the decoder (450), a buffer receives encoded data for a compressed frame and, at an appropriate time, makes the received encoded data available to an entropy decoder. The entropy decoder entropy decodes entropy-coded quantized data as well as entropy-coded side information, typically applying the inverse of entropy encoding performed in the encoder. A motion compensator applies motion information to one or more reference frames to form motion-compensated prediction values for any inter-coded blocks of the frame being reconstructed. An intra-frame prediction module can spatially predict sample values of a current block from neighboring, previously reconstructed sample values or, for intra BC prediction, predict sample values of a current block using previously reconstructed sample values of an intra-frame prediction region in the frame. The intra-frame prediction region can be indicated with a BV value. The decoder (450) also reconstructs prediction residual values. An inverse quantizer inverse quantizes entropy-decoded data. For example, the decoder (450) sets values for QP for a picture, tile, slice and/or other portion of video based on syntax elements in the bitstream, and inverse quantizes transform coefficients accordingly. An inverse frequency transformer converts the quantized, frequency-domain data into spatial-domain data. For an inter-frame predicted block, the decoder (450) combines reconstructed prediction residual values with motion-compensated prediction values. The decoder (450) can similarly combine prediction residual

values with prediction values from intra prediction. An adaptive deblocking filter is included within the motion compensation loop in the video decoder (450) to smooth discontinuities across block boundary rows and/or columns in the decoded frame (451). Other filtering (such as de-ringing filtering, ALF, or SAO filtering; not shown) can alternatively or additionally be applied as in-loop filtering operations.

[074] The decoded frame temporary memory storage area (460) includes multiple frame buffer storage areas (461, 462, ..., 46*n*). The decoded frame storage area (460) is an example of a decoded picture buffer. The decoder (450) uses the MMCO/RPS information (432) to identify a frame buffer (461, 462, etc.) in which it can store a decoded frame (451). The decoder (450) stores the decoded frame (451) in that frame buffer.

[075] An output sequencer (480) identifies when the next frame to be produced in output order is available in the decoded frame storage area (460). When the next frame (481) to be produced in output order is available in the decoded frame storage area (460), it is read by the output sequencer (480) and output to the output destination (490) (*e.g.*, display). In general, the order in which frames are output from the decoded frame storage area (460) by the output sequencer (480) may differ from the order in which the frames are decoded by the decoder (450).

V. Example Video Encoders.

[076] Figures 5a and 5b are a block diagram of a generalized video encoder (500) in conjunction with which some described embodiments may be implemented. The encoder (500) receives a sequence of video pictures including a current picture as an input video signal (505) and produces encoded data in a coded video bitstream (595) as output.

[077] The encoder (500) is block-based and uses a block format that depends on implementation. Blocks may be further sub-divided at different stages, *e.g.*, at the prediction, frequency transform and/or entropy encoding stages. For example, a picture can be divided into 64x64 blocks, 32x32 blocks or 16x16 blocks, which can in turn be divided into smaller blocks of sample values for coding and decoding. In implementations of encoding for the

HEVC standard, the encoder partitions a picture into CTUs (CTBs), CUs (CBs), PUs (PBs) and TU (TBs).

[078] The encoder (500) compresses pictures using intra-picture coding and/or inter-picture coding. Many of the components of the encoder (500) are used for both intra-picture coding and inter-picture coding. The exact operations performed by those components can vary depending on the type of information being compressed.

[079] A tiling module (510) optionally partitions a picture into multiple tiles of the same size or different sizes. For example, the tiling module (510) splits the picture along tile rows and tile columns that, with picture boundaries, define horizontal and vertical boundaries of tiles within the picture, where each tile is a rectangular region.

[080] The general encoding control (520) receives pictures for the input video signal (505) as well as feedback (not shown) from various modules of the encoder (500). Overall, the general encoding control (520) provides control signals (not shown) to other modules (such as the tiling module (510), transformer/scaler/quantizer (530), scaler/inverse transformer (535), intra-picture estimator (540), motion estimator (550) and intra/inter switch) to set and change coding parameters during encoding. In particular, the general encoding control (520) can decide whether and how to use aspects of BV prediction (*e.g.*, skip mode with predicted BV values, merge mode for BV prediction) during encoding. The general encoding control (520) can also evaluate intermediate results during encoding, for example, performing rate-distortion analysis. The general encoding control (520) produces general control data (522) that indicates decisions made during encoding, so that a corresponding decoder can make consistent decisions. The general control data (522) is provided to the header formatter/entropy coder (590).

[081] If the current picture is predicted using inter-picture prediction, a motion estimator (550) estimates the motion of blocks of sample values of the current picture of the input video signal (505) with respect to one or more reference pictures. The decoded picture buffer (570) buffers one or more reconstructed previously coded pictures for use as reference pictures. When multiple reference pictures are used, the multiple reference pictures can be from different temporal directions or the same temporal direction. The motion estimator (550)

produces as side information motion data (552) such as MV data, merge mode index values, and reference picture selection data. The motion data (552) is provided to the header formatter/entropy coder (590) as well as the motion compensator (555).

[082] The motion compensator (555) applies MVs to the reconstructed reference picture(s) from the decoded picture buffer (570). The motion compensator (555) produces motion-compensated predictions for the current picture.

[083] In a separate path within the encoder (500), an intra-picture estimator (540) determines how to perform intra-picture prediction for blocks of sample values of a current picture of the input video signal (505). The current picture can be entirely or partially coded using intra-picture coding. Using values of a reconstruction (538) of the current picture, for intra spatial prediction, the intra-picture estimator (540) determines how to spatially predict sample values of a current block of the current picture from neighboring, previously reconstructed sample values of the current picture. Or, for intra BC prediction using BV values, the intra-picture estimator (540) estimates displacement of the sample values of the current block to different candidate regions within the current picture.

[084] The intra-picture estimator (540) produces as side information intra prediction data (542), such as information indicating whether intra prediction uses spatial prediction or intra BC prediction (*e.g.*, a flag value per intra block), prediction mode direction (for intra spatial prediction), and BV values (for intra BC prediction). The intra prediction data (542) is provided to the header formatter/entropy coder (590) as well as the intra-picture predictor (545).

[085] According to the intra prediction data (542), the intra-picture predictor (545) spatially predicts sample values of a current block of the current picture from neighboring, previously reconstructed sample values of the current picture. Or, for intra BC prediction, the intra-picture predictor (545) predicts the sample values of the current block using previously reconstructed sample values of an intra-frame prediction region, which is indicated by a BV value for the current block. In some cases (*e.g.*, for a skip-mode block or merge mode block), the BV value can be a BV predictor (predicted BV value). When the chroma data for a picture has the same resolution as the luma data (*e.g.* when the format is YUV 4:4:4 format or

RGB 4:4:4 format), the BV value that is applied for the chroma block may be the same as the BV value applied for the luma block. On the other hand, when the chroma data for a picture has reduced resolution relative to the luma data (*e.g.* when the format is YUV 4:2:0 format), the BV value that is applied for the chroma block may be scaled down and possibly rounded to adjust for the difference in chroma resolution (*e.g.* by dividing the vertical and horizontal components of the BV value by two and truncating or rounding them to integer values).

[086] The intra/inter switch selects values of a motion-compensated prediction or intra-picture prediction for use as the prediction (558) for a given block. The difference (if any) between a block of the prediction (558) and a corresponding part of the original current picture of the input video signal (505) provides values of the residual (518), for a non-skip-mode block. During reconstruction of the current picture, for a non-skip-mode block, reconstructed residual values are combined with the prediction (558) to produce a reconstruction (538) of the original content from the video signal (505). In lossy compression, however, some information is still lost from the video signal (505).

[087] In the transformer/scaler/quantizer (530), a frequency transformer converts spatial-domain video information into frequency-domain (*i.e.*, spectral, transform) data. For block-based video coding, the frequency transformer applies a discrete cosine transform ("DCT"), an integer approximation thereof, or another type of forward block transform (*e.g.*, a discrete sine transform or an integer approximation thereof) to blocks of prediction residual data (or sample value data if the prediction (558) is null), producing blocks of frequency transform coefficients. The encoder (500) may also be able to indicate that such transform step is skipped. The scaler/quantizer scales and quantizes the transform coefficients. For example, the quantizer applies dead-zone scalar quantization to the frequency-domain data with a quantization step size that varies on a frame-by-frame basis, tile-by-tile basis, slice-by-slice basis, block-by-block basis, frequency-specific basis or other basis. The quantized transform coefficient data (532) is provided to the header formatter/entropy coder (590).

[088] In the scaler/inverse transformer (535), a scaler/inverse quantizer performs inverse scaling and inverse quantization on the quantized transform coefficients. An inverse frequency transformer performs an inverse frequency transform, producing blocks of reconstructed prediction residual values or sample values. For a non-skip-mode block, the

encoder (500) combines reconstructed residual values with values of the prediction (558) (*e.g.*, motion-compensated prediction values, intra-picture prediction values) to form the reconstruction (538). For a skip-mode block, the encoder (500) uses the values of the prediction (558) as the reconstruction (538).

[089] For intra-picture prediction, the values of the reconstruction (538) can be fed back to the intra-picture estimator (540) and intra-picture predictor (545). Also, the values of the reconstruction (538) can be used for motion-compensated prediction of subsequent pictures. The values of the reconstruction (538) can be further filtered. A filtering control (560) determines how to perform deblock filtering and SAO filtering on values of the reconstruction (538), for a given picture of the video signal (505). The filtering control (560) produces filter control data (562), which is provided to the header formatter/entropy coder (590) and merger/filter(s) (565).

[090] In the merger/filter(s) (565), the encoder (500) merges content from different tiles into a reconstructed version of the picture. The encoder (500) selectively performs deblock filtering and SAO filtering according to the filter control data (562), so as to adaptively smooth discontinuities across boundaries in the frames. Other filtering (such as de-ringing filtering or ALF; not shown) can alternatively or additionally be applied. Tile boundaries can be selectively filtered or not filtered at all, depending on settings of the encoder (500), and the encoder (500) may provide syntax within the coded bitstream to indicate whether or not such filtering was applied. The decoded picture buffer (570) buffers the reconstructed current picture for use in subsequent motion-compensated prediction.

[091] The header formatter/entropy coder (590) formats and/or entropy codes the general control data (522), quantized transform coefficient data (532), intra prediction data (542), motion data (552) and filter control data (562). For the intra prediction data (542), the header formatter/entropy coder (590) selects and entropy codes BV predictor index values (for intra BC prediction), *e.g.*, using context-adaptive binary arithmetic coding. For non-skip-mode, non-merge-mode blocks, the header formatter/entropy coder (590) determines BV differentials for BV values (relative to BV predictors for the BV values), then entropy codes the BV differentials, *e.g.*, using context-adaptive binary arithmetic coding. For a skip-mode block or merge-mode block, the BV differential is omitted.

[092] The header formatter/entropy coder (590) provides the encoded data in the coded video bitstream (595). The format of the coded video bitstream (595) can be a variation or extension of HEVC format, Windows Media Video format, VC-1 format, MPEG-x format (*e.g.*, MPEG-1, MPEG-2, or MPEG-4), H.26x format (*e.g.*, H.261, H.262, H.263, H.264), or another format.

[093] Depending on implementation and the type of compression desired, modules of the encoder can be added, omitted, split into multiple modules, combined with other modules, and/or replaced with like modules. In alternative embodiments, encoders with different modules and/or other configurations of modules perform one or more of the described techniques. Specific embodiments of encoders typically use a variation or supplemented version of the encoder (500). The relationships shown between modules within the encoder (500) indicate general flows of information in the encoder; other relationships are not shown for the sake of simplicity.

VI. Example Video Decoders.

[094] Figure 6 is a block diagram of a generalized decoder (600) in conjunction with which some described embodiments may be implemented. The decoder (600) receives encoded data in a coded video bitstream (605) and produces output including pictures for reconstructed video (695). The format of the coded video bitstream (605) can be a variation or extension of HEVC format, Windows Media Video format, VC-1 format, MPEG-x format (*e.g.*, MPEG-1, MPEG-2, or MPEG-4), H.26x format (*e.g.*, H.261, H.262, H.263, H.264), or another format.

[095] The decoder (600) is block-based and uses a block format that depends on implementation. Blocks may be further sub-divided at different stages. For example, a picture can be divided into 64x64 blocks, 32x32 blocks or 16x16 blocks, which can in turn be divided into smaller blocks of sample values. In implementations of decoding for the HEVC standard, a picture is partitioned into CTUs (CTBs), CUs (CBs), PUs (PBs) and TU (TBs).

[096] The decoder (600) decompresses pictures using intra-picture decoding and/or inter-picture decoding. Many of the components of the decoder (600) are used for both intra-picture decoding and inter-picture decoding. The exact operations performed by those components can vary depending on the type of information being decompressed.

[097] A buffer receives encoded data in the coded video bitstream (605) and makes the received encoded data available to the parser/entropy decoder (610). The parser/entropy decoder (610) entropy decodes entropy-coded data, typically applying the inverse of entropy coding performed in the encoder (500) (*e.g.*, context-adaptive binary arithmetic decoding). As a result of parsing and entropy decoding, the parser/entropy decoder (610) produces general control data (622), quantized transform coefficient data (632), intra prediction data (642), motion data (652) and filter control data (662). For the intra prediction data (642), the parser/entropy decoder (610) entropy decodes BV predictor index values (for intra BC prediction), *e.g.*, using context-adaptive binary arithmetic decoding. For non-skip-mode, non-merge-mode blocks, the parser/entropy decoder (610) also entropy decodes BV differentials for BV values (*e.g.*, using context-adaptive binary arithmetic decoding), then combines the BV differentials with corresponding BV predictors to reconstruct the BV values. For a skip-mode block or merge-mode block, the BV differential is omitted from the bitstream, and the BV value is simply the BV predictor (*e.g.*, indicated with the BV predictor index value).

[098] The general decoding control (620) receives the general control data (622) and provides control signals (not shown) to other modules (such as the scaler/inverse transformer (635), intra-picture predictor (645), motion compensator (655) and intra/inter switch) to set and change decoding parameters during decoding.

[099] If the current picture is predicted using inter-picture prediction, a motion compensator (655) receives the motion data (652), such as MV data, reference picture selection data and merge mode index values. The motion compensator (655) applies MVs to the reconstructed reference picture(s) from the decoded picture buffer (670). The motion compensator (655) produces motion-compensated predictions for inter-coded blocks of the current picture. The decoded picture buffer (670) stores one or more previously reconstructed pictures for use as reference pictures.

[0100] In a separate path within the decoder (600), the intra-frame prediction predictor (645) receives the intra prediction data (642), such as information indicating whether intra prediction uses spatial prediction or intra BC prediction (*e.g.*, a flag value per intra block), prediction mode direction (for intra spatial prediction) and BV values (for intra BC prediction). For intra spatial prediction, using values of a reconstruction (638) of the current picture, according to prediction mode data, the intra-picture predictor (645) spatially predicts sample values of a current block of the current picture from neighboring, previously reconstructed sample values of the current picture. Or, for intra BC prediction using BV values, the intra-picture predictor (645) predicts the sample values of the current block using previously reconstructed sample values of an intra-frame prediction region, which is indicated by a BV value for the current block.

[0101] The intra/inter switch selects values of a motion-compensated prediction or intra-picture prediction for use as the prediction (658) for a given block. For example, when HEVC syntax is followed, the intra/inter switch can be controlled based on a syntax element encoded for a CU of a picture that can contain intra-predicted CUs and inter-predicted CUs. For a non-skip-mode block, the decoder (600) combines the prediction (658) with reconstructed residual values to produce the reconstruction (638) of the content from the video signal. For a skip-mode block, the decoder (600) uses the values of the prediction (658) as the reconstruction (638).

[0102] To reconstruct the residual for a non-skip-mode block, the scaler/inverse transformer (635) receives and processes the quantized transform coefficient data (632). In the scaler/inverse transformer (635), a scaler/inverse quantizer performs inverse scaling and inverse quantization on the quantized transform coefficients. An inverse frequency transformer performs an inverse frequency transform, producing blocks of reconstructed prediction residual values or sample values. For example, the inverse frequency transformer applies an inverse block transform to frequency transform coefficients, producing sample value data or prediction residual data. The inverse frequency transform can be an inverse DCT, an integer approximation thereof, or another type of inverse frequency transform (*e.g.*, an inverse discrete sine transform or an integer approximation thereof).

[0103] For intra-picture prediction, the values of the reconstruction (638) can be fed back to the intra-picture predictor (645). For inter-picture prediction, the values of the reconstruction (638) can be further filtered. In the merger/filter(s) (665), the decoder (600) merges content from different tiles into a reconstructed version of the picture. The decoder (600) selectively performs deblock filtering and SAO filtering according to the filter control data (662) and rules for filter adaptation, so as to adaptively smooth discontinuities across boundaries in the frames. Other filtering (such as de-ringing filtering or ALF; not shown) can alternatively or additionally be applied. Tile boundaries can be selectively filtered or not filtered at all, depending on settings of the decoder (600) or a syntax indication within the encoded bitstream data. The decoded picture buffer (670) buffers the reconstructed current picture for use in subsequent motion-compensated prediction.

[0104] The decoder (600) can also include a post-processing filter. The post-processing filter (608) can include de-ringing filtering, adaptive Wiener filtering, film-grain reproduction filtering, SAO filtering or another kind of filtering.

[0105] Depending on implementation and the type of decompression desired, modules of the decoder can be added, omitted, split into multiple modules, combined with other modules, and/or replaced with like modules. In alternative embodiments, decoders with different modules and/or other configurations of modules perform one or more of the described techniques. Specific embodiments of decoders typically use a variation or supplemented version of the decoder (600). The relationships shown between modules within the decoder (600) indicate general flows of information in the decoder; other relationships are not shown for the sake of simplicity.

VII. Innovations in Block Vector Prediction.

[0106] This section presents various features of block vector ("BV") prediction. Some of the features relate to use of a default BV predictor, while other features relate to merge mode for BV predictors or skip mode for intra block copy ("BC") prediction. These features can facilitate intra BC prediction that is more effective in terms of rate-distortion performance.

[0107] In particular, the described innovations can improve rate-distortion performance when encoding certain "artificially" created video content such as screen-capture content. Screen-capture content typically includes repeated structures (*e.g.*, graphics, text characters), which provide opportunities for intra BC prediction to improve performance. Screen capture content is usually encoded in a format (*e.g.*, YUV 4:4:4 or RGB 4:4:4) with high chroma sampling resolution, although it may also be encoded in a format with lower chroma sampling resolution (*e.g.,* YUV 4:2:0). Common scenarios for encoding/decoding of screen-capture content include remote desktop conferencing and encoding/decoding of graphical overlays on natural video or other "mixed content" video.

A. Intra BC Prediction Mode, BV Values and BV Prediction – Introduction.

[0108] For intra BC prediction, the sample values of a current block of a picture are predicted using sample values in the same picture. A BV value indicates a displacement from the current block to a region of the picture that includes the sample values used for prediction. The sample values used for prediction are previously reconstructed sample values. The BV value can be signaled in the bitstream, and a decoder can use the BV value to determine the region of the picture to use for prediction, which is also reconstructed at the decoder. Intra BC prediction is a form of intra-picture prediction – intra BC prediction for a block of a picture does not use any sample values other than sample values in the same picture.

[0109] Figure 7a illustrates intra BC prediction for a current block (760) of a current frame (710). The current block can be a coding block ("CB") of a coding unit ("CU"), prediction block ("PB") of a prediction unit ("PU"), transform block ("TB") of a transform unit ("TU") or other block. The size of the current block can be 64x64, 32x32, 16x16, 8x8 or some other size. More generally, the size of the current block is m x n, where each of m and n is a whole number, and where m and n can be equal to each other or can have different values. Alternatively, the current block can have some other shape (*e.g.*, an area of a coded video object with a non-rectangular shape).

[0110] The BV (761) indicates a displacement (or offset) from the current block (760) to a region (762) of the picture that includes the sample values used for prediction. The intra-frame prediction region (762) indicated by the BV (761) is sometimes termed the "matching

block" for the current block (760). The matching block can be identical to the current block (760), or it can be an approximation of the current block (760). Suppose the top left position of a current block is at position (x_0, y_0) in the current frame, and suppose the top left position of the intra-frame prediction region is at position (x_1, y_1) in the current frame. The BV indicates the displacement $(x_1 - x_0, y_1 - y_0)$. For example, if the top left position of the current block is at position (256, 128), and the top left position of the intra-frame prediction region is at position (176, 104), the BV value is (-80, -24). In this example, a negative horizontal displacement indicates a position to the left of the current block, and a negative vertical displacement indicates a position above the current block.

[0111] Intra BC prediction can improve coding efficiency by exploiting redundancy (such as repeated patterns inside a frame) using BC operations. The sample values of a current block are represented using a BV value instead of directly encoding the sample values of the current block. Even if the sample values of the current block do not exactly match the sample values of the intra-frame prediction region indicated with the BV value, the differences may be negligible (not perceptually noticeable). Or, if the differences are significant, the differences may be encoded as residual values that can be compressed more efficiently than the original sample values for the current block.

[0112] Collectively, BV values for blocks encoded using intra BC prediction can consume a significant number of bits. The BV values can be entropy encoded to reduce bit rate. To further reduce bit rate for BV values, an encoder can use prediction of the BV values. BV values often exhibit redundancy – the BV value for a given block is often similar to, or even the same as, the BV values of previous blocks in the picture. For BV prediction, the BV value for the given block is predicted using a BV predictor. The difference (or BV differential) between the BV value for the given block and the BV predictor is then entropy coded. Typically, the BV differential is computed for horizontal and vertical components of the BV value and BV predictor. When BV prediction works well, BV differentials have a probability distribution that supports efficient entropy coding. In the current draft version of the HEVC standard (JCTVC-O1005), the BV predictor is the BV value of the last coded CU within the current CTU (that is, the BV value of the previous intra-BC-predicted block within the current CTU).

[0113] Figure 7b shows a BV (761) of a current block (760) in a frame as well as a BV (751) of a previous block (750) in the frame (710). The BV (751) of the previous block (750) is used as the BV predictor for the BV (761) of the current block (760). For example, if the BV value is (-80, -24) and the BV predictor is (-80, -32), the BV differential of (0, 8) is entropy encoded.

[0114] A decoder receives and entropy decodes the entropy coded BV differential for a BV value. The decoder also determines a BV predictor for the BV value. The BV predictor determined by the decoder is the same as the BV predictor determined by the encoder. The decoder combines the BV predictor and decoded BV differential to reconstruct the BV value.

[0115] In the current draft version of the HEVC standard (JCTVC-O1005_v3), a BV predictor based on an actual BV value is not available in some situations. For example, a previous actual BV value is not available for the first intra BC-predicted block in a given CTU. When a previous actual BV value is not available for the BV predictor (as shown in Figure 7c), a default value of (0, 0) is used as the BV predictor. In effect, this means BV prediction is skipped when no BV value is available for a previous intra-BC-predicted block in the CTU.

B. Default BV Predictor with Non-zero Component.

[0116] According to one aspect of the innovations described herein, when a previous actual BV value is not available for use as a BV predictor for the BV value of a current block, an encoder and decoder use a default BV predictor with a non-zero component. Compared to a zero-value default BV predictor, a default BV predictor with a non-zero component tends to be closer to the BV value of the current block, which results more efficient entropy coding of the BV differential. Also, a default BV predictor with a non-zero value can provide a useful option for skip mode or merge mode for BV prediction.

1. Example Default BV Predictors.

[0117] As shown in Figure 7d, for the BV value (761) of the current block (760), the default BV predictor (763) can have a non-zero horizontal component. For example, the default BV predictor is a horizontal vector (BVx, 0). The value of the horizontal BV

component BVx can be a fixed non-zero value. Or, the value of BVx can be the width of the current block, which is the smallest value of horizontal BV component BVx that avoids overlap between the current block and the reference region.

[0118] For example, when the current block is a CB of a CU having width *W* and height *H*, the value of the horizontal BV component BVx for the default BV predictor can be a negative offset equal to the width *W* of the CU. That is, the default BV predictor can be (-*W*, 0). The default BV predictor (BVx, 0) can be used whenever a regular BV predictor (based on an actual BV value of a previous block) is unavailable. Or, the default BV predictor (BVx, 0) can be conditionally used depending on the BV value of the current block. For example, if the vertical BV component of the BV value of the current block is less than -*H*, the default BV predictor is (-*W*, 0). Otherwise (the vertical BV component of the BV value of the current block is not less than -*H*), the default BV predictor is (0, 0). This condition ensures that the default BV predictor of (-*W*, 0) is not applied when the BV value of the current block has a significant vertical BV component. Alternatively, the non-zero default BV predictor (BVx, 0) is conditionally applied according to another condition.

[0119] Instead of having a negative offset of -*W*, the default BV predictor can have another negative offset. For example, the default BV predictor can be (-2**W*, 0).

[0120] Alternatively, the default BV predictor is a vertical vector (0, BVy). The value of the vertical BV component BVy can be a fixed non-zero value, or the value of BVy can be the height of the current block, which is the smallest value of BVy that avoids overlap between the current block and the reference region. For example, when the current block is a CB of a CU having width *W* and height *H*, the default BV predictor can be (0, -*H*). The default BV predictor (0, BVy) can be used whenever a regular BV predictor is unavailable, or the default BV predictor (0, BVy) can be conditionally used depending on the BV value of the current block.

[0121] Alternatively, the default BV predictor can have a non-zero horizontal component and non-zero vertical component.

2. Example Encoding with Default BV Predictors.

[0122] Figure 8 shows a generalized technique (800) for using a default BV predictor with a non-zero component during encoding. An image encoder or video encoder such as one described with reference to Figure 3 or Figures 5a-5b can perform the technique (800).

[0123] The encoder determines (830) a default BV predictor for a current block of a picture. For example, in HEVC implementations, the current block is a part of a coding unit of a coding tree unit. The default BV predictor includes a BV predictor component with a non-zero value. For example, the BV predictor component with a non-zero value is a horizontal BV component. Depending on implementation, the non-zero horizontal BV component can have a fixed value (independent of the size of the current block), a value equal to the width of the current block, or some other non-zero value. Or, the BV predictor component with a non-zero value is a vertical BV component. Depending on implementation, the non-zero vertical BV component can have a fixed value (independent of the size of the current block), a value equal to the height of the current block, or some other non-zero value. Or, the default BV predictor can include a non-zero horizontal BV component and a non-zero vertical BV component. The default BV predictor can be part of a set of multiple BV predictor candidates assembled for skip mode, as described in section C, or for merge mode, as described in section D.

[0124] The encoder encodes (850) the current block using the default BV predictor. For example, the encoder (a) performs intra BC prediction using a BV value for the current block, (b) determines a BV differential using the BV value and the default BV predictor for the current block, and (c) encodes the BV differential. Alternatively, the encoder simply performs intra BC prediction using the default BV predictor.

[0125] Figure 9 shows a more detailed example technique (900) for using a default BV predictor with a non-zero component during encoding. An image encoder or video encoder such as one described with reference to Figure 3 or Figures 5a-5b can perform the technique (900).

[0126] To start, the encoder identifies (910) a BV value for a current block of a picture. The encoder can use any form of BV estimation to identify the BV value for the current block. In HEVC implementations, the current block is a part of a coding unit of a coding tree unit.

[0127] The encoder checks (920) whether an actual BV is available for use as BV predictor. For example, the encoder checks whether an actual BV value for a previous block of the picture is available. In HEVC implementations, an actual BV value for a previous block is available if (1) the previous block and the current block are part of the same CTU, and (2) the prediction mode of the previous block is intra BC prediction mode. An actual BV value for a previous block is not available, for example, for the first intra-BC-predicted block of a given CTU.

[0128] If an actual BV is available for use as BV predictor, the encoder uses (940) the actual BV as the BV predictor. On the other hand, if no actual BV for any previous block is available for use as BV predictor, the encoder determines (930) a default BV predictor for the current block. The default BV predictor includes a BV predictor component with a non-zero value. For example, the default BV predictor is determined as described above with respect to act (830) of Figure 8.

[0129] The encoder determines (950) a BV differential between the BV value for the current block and the BV predictor. The encoder entropy codes (960) the BV differential.

3. Example Decoding with Default BV Predictors.

[0130] Figure 10 shows a generalized technique (1000) for using a default BV predictor with a non-zero component during decoding. An image decoder or video decoder such as one described with reference to Figure 4 or Figure 6 can perform the technique (1000).

[0131] The decoder determines (1030) a default BV predictor for a current block of a picture. For example, in HEVC implementations, the current block is a part of a coding unit of a coding tree unit. The default BV predictor includes a BV predictor component with a non-zero value. For example, the BV predictor component with a non-zero value is a horizontal BV component. Depending on implementation, the non-zero horizontal BV component can have a fixed value (independent of the size of the current block), a value equal to the width of the current block, or some other non-zero value. Or, the BV predictor component with a non-zero value is a vertical BV component. Depending on implementation, the non-zero vertical BV component can have a fixed value (independent of the size of the current block), a value equal to the height of the current block, or some other non-zero value.

Or, the default BV predictor can include a non-zero horizontal BV component and a non-zero vertical BV component. The default BV predictor can be part of a set of multiple BV predictor candidates assembled for skip mode, as described in section C, or for merge mode, as described in section D.

[0132] The decoder decodes (1050) the current block using the default BV predictor. For example, the encoder (a) decodes a BV differential for the current block, (b) combines the decoded BV differential with the default BV predictor to reconstruct a BV value for the current block, and (c) performs intra BC prediction using the BV value for the current block. Alternatively, the decoder simply performs intra BC prediction using the default BV predictor.

[0133] Figure 11 shows a more detailed example technique (1100) for using a default BV predictor with a non-zero component during decoding. An image decoder or video decoder such as one described with reference to Figure 4 or Figure 6 can perform the technique (1100).

[0134] The decoder checks (1120) whether an actual BV is available for use as BV predictor for a BV value for a current block of a picture. For example, the decoder checks whether an actual BV value for a previous block of the picture is available. In HEVC implementations, an actual BV value for a previous block is available if (1) the previous block and the current block are part of the same CTU, and (2) the prediction mode of the previous block is intra BC prediction mode. An actual BV value for a previous block is not available, for example, for the first intra-BC-predicted block of a given CTU.

[0135] If an actual BV is available for use as BV predictor, the decoder uses (1140) the actual BV as the BV predictor. On the other hand, if no actual BV for any previous block is available for use as BV predictor, the decoder determines (1130) a default BV predictor for the current block. The default BV predictor includes a BV predictor component with a non-zero value. For example, the default BV predictor is determined as described above with respect to act (1030) of Figure 10.

[0136] The decoder also entropy decodes (1150) a BV differential for the current block. The decoder then combines (1160) the BV differential with the BV predictor to reconstruct the BV value for the current block.

4. Alternatives and Variations.

[0137] In the examples of default BV predictors described above, the value of the default BV predictor is not constrained depending on location of the current block. Alternatively, the value of the default BV predictor can be constrained depending on the location of the current block. For example, if the current block is at the edge of a picture, such that the default BV predictor would point to a location outside the picture, the default BV predictor is constrained to point to a location within the picture, or a (0, 0) BV predictor is used. Or, as another example for HEVC implementations in which BV values are constrained reference a location in some set of CTUs (*e.g.*, the current CTU and previous CTU), the default BV predictor is constrained to reference a location within that set of CTUs. Or, as another example for HEVC implementations in which a picture includes multiple tiles or multiple slices, BV values (and default BV predictors) may be constrained to reference a location with the current tile (and/or current slice).

[0138] In the examples of default BV predictors described above, the value of the default BV predictor does not switch depending on location of the current block. Alternatively, the value of the default BV predictor can switch depending on the location of the current block. For example, the default BV predictor is a horizontal BV predictor (BVx, 0) if the current block is anywhere except the left edge of the picture (or tile, or slice, if used). If the current block is at the left edge of the picture (or tile, or slice, if used), or the default horizontal BV predictor would otherwise reference a location outside the picture (or tile, or slice, if used), the default BV predictor is a vertical BV predictor (0, BVy). And, if the current block is at the top left corner of the picture (or tile, or slice, if used), or the default horizontal BV predictor and vertical BV predictor would otherwise reference a location outside the picture (or tile, or slice, if used), the default BV predictor is a (0, 0) BV predictor.

C. Skip Mode for Intra BC Prediction.

[0139] According to another aspect of the innovations described herein, an encoder and decoder use skip mode for certain intra-BC-predicted blocks. In skip mode, an intra-BC-predicted block uses a predicted BV value and has no residual data in the bitstream. In some scenarios, use of skip mode for intra-BC-predicted blocks can significantly improve rate-distortion efficiency.

[0140] In general, an encoder decides when to use skip mode for a current intra-BC-predicted block. The encoder can decide to use skip mode based on how closely an intra-BC-prediction region matches a current block, based on the perceptual importance of residual values, based on the bitrate available for encoding of residual values, and/or based on other factors. The encoder can favor skip mode for intra-BC-predicted blocks in certain encoding modes in which encoding speed is a constraint, *e.g.*, real-time encoding.

[0141] Figure 12 shows a generalized technique (1200) for encoding that includes skip mode for intra-BC-predicted blocks. An image encoder or video encoder such as one described with reference to Figure 3 or Figures 5a-5b can perform the technique (1200).

[0142] For a current block of a picture, if the current block is an intra-BC-predicted block encoded in skip mode, the encoder uses (1210) BV prediction to determine a BV value for the current block. The BV value for the current block indicates a displacement to a region within the picture. The BV prediction can include merge mode prediction, prediction using the BV value of the last coded intra-BC-predicted block, prediction using a component-wise median of BV values of neighboring intra-BC-predicted blocks, or some other form of BV prediction. The current block is, for example, a part of a CU of a CTU.

[0143] The encoder encodes (1220) the current block using intra BC prediction with the BV value (here, the predicted BV value). The encoder outputs (1230) in a bitstream an indication that the current block is encoded in skip mode. Since the block is encoded in skip mode, the bitstream lacks a BV differential and residual data for the current block. The bitstream can include, however, an index value that indicates a selection of a BV predictor candidate, from a set of multiple BV predictor candidates, to use as the BV value for the current block. For example, when the set of multiple BV predictor candidates has two BV predictor candidates, the index value is a flag value. Or, when the set of multiple BV

predictor candidates has more than two BV predictor candidates, the index value is an integer value. The index value can be entropy coded, *e.g.*, using arithmetic coding.

[0144] Figure 13 shows a generalized technique (1300) for decoding that includes skip mode for intra-BC-predicted blocks. An image decoder or video decoder such as one described with reference to Figure 4 or Figure 6 can perform the technique (1300).

[0145] For a current block of a picture, if the current block is an intra-BC-predicted block encoded in skip mode, the decoder receives (1310) from a bitstream an indication that the current block is encoded in skip mode. Since the block is encoded in skip mode, the bitstream lacks a BV differential and residual data for the current block. As described above, however, the bitstream can include an index value that indicates a selection of a BV predictor candidate, from a set of multiple BV predictor candidates, to use as the BV value for the current block. The current block is, for example, a part of a CU of a CTU.

[0146] The decoder uses (1320) BV prediction to determine a BV value for the current block. The BV prediction generally mirrors the BV prediction performed at the encoder. It can include merge mode prediction, prediction using the BV value of the last coded intra-BC-predicted block, prediction using a component-wise median of BV values of neighboring intra-BC-predicted blocks, or some other form of BV prediction. The decoder decodes (1330) the current block using intra BC prediction with the BV value (here, the predicted BV value).

D. Merge Mode for BV Prediction.

[0147] According to another aspect of the innovations described herein, an encoder and decoder use merge mode for BV prediction. Merge mode can significantly improve the quality of BV prediction, which in turn reduces the amount of bits consumed by BV differentials in the bitstream.

[0148] For merge mode, when determining a BV predictor for a block, an encoder assembles a set of BV predictor candidates, selects one of the BV predictor candidates to use as the BV predictor for the block, and signals in the bitstream an index value that indicates the selected BV predictor candidate. A decoder assembles the set of BV predictor candidates and, based on the signaled index value, selects one of the BV predictor candidates to use as

the BV predictor for the block. Although the index values consume bits in the bitstream, merge mode can significantly improve BV prediction and hence reduce bits consumed by BV differentials, resulting in an overall reduction in bitrate.

[0149] In example implementations, merge mode for BV prediction is used in combination with skip mode for intra-BC-predicted blocks. Alternatively, merge mode for BV prediction can be used even if skip mode is not enabled for intra-BC-predicted blocks.

1. Example BV Predictor Candidates and Index Values.

[0150] When determining a BV predictor for a block in merge mode, an encoder or decoder assembles a set of BV predictor candidates. The constituents of the set of BV predictor candidates depend on implementation, but are the same at the encoder side and the decoder side.

[0151] In one approach, BV predictor candidates are assembled from among actual BV values of previously coded blocks in a picture. In general, the encoder and decoder determine up to x actual BV values of previous blocks in the picture. The previous blocks are previously coded intra-BC-predicted blocks in the picture. In particular, they are the x closest intra-BC-predicted blocks earlier in decoding order than the current block. The value of x depends on implementation and can be, for example, 2, 3, 4, 5 or some other number. Using more BV predictor candidates tends to increase bit rate for index values but improve the quality of BV prediction, which in turn reduces bits consumed by BV differentials.

[0152] The previous intra-BC-predicted blocks need not be neighbors of the current block. If less than x previous blocks with actual BV values are available, the set of BV predictor candidates can include less than x BV predictor candidates.

[0153] Figure 14 shows an example of this approach. In Figure 14, three BV values (1431, 1441, 1451) of three previous intra-BC-predicted blocks (1430, 1440, 1450) provide the BV predictor candidates for merge mode for the BV predictor of a current block (1460) of a current frame (1410). Although the previous blocks (1430, 1440, 1450) are contiguous in the example of Figure 14, the previous blocks need not be contiguous. (That is, the previous

intra-BC-predicted blocks that provide BV predictor candidates can be separated by other, non-intra-BC-predicted blocks in the picture.)

[0154] The set of BV predictor candidates can further include one or more default BV predictors. In the example of Figure 14, the set of BV predictor candidates includes a default BV predictor (BVx, 0) with horizontal displacement and a default BV predictor (0, BVy) with vertical displacement. In particular, default BV predictors with horizontal displacement and vertical displacement tend to be useful for text content, which generally follows a pure horizontal or vertical direction.

[0155] Merge mode index values are associated with the respective BV predictor candidates. Typically, smaller index values are associated with BV predictor candidates that are more likely to be selected, and larger index values are associated with BV predictor candidates that are less likely to be selected, which tends to reduce bit rate for the index values after entropy coding. In Figure 14, index values idx 0 ... idx 4 are assigned to the BV predictor candidates with BV values (1451, 1441, 1431) and default BV predictors according to expected likelihood of selection.

[0156] For example, in some implementations, two actual BV values of previously coded blocks are stored as BV predictor candidates BV_{cand0} and BV_{cand1}. After a new BV value has been used, the older BV predictor candidate BV_{cand1} is replaced with the newer BV predictor candidate BV_{cand0}, and the newer BV predictor candidate BV_{cand0} is replaced with the BV value that was just used. At the encoder side, for a current block, the encoder selects one of BV_{cand0} and BV_{cand1} to use as BV predictor. As a merge mode index value, the encoder signals a flag value that indicates which one of BV_{cand0} and BV_{cand1} has been used by the encoder and should be used by the decoder. At the decoder side, a decoder receives and parses the flag value and, based on the flag value, selects one of BV_{cand0} and BV_{cand1} to use as the BV predictor for the block. In this approach, the encoder and the decoder each store and update two BV predictor candidates BV_{cand0} and BV_{cand1}.

[0157] As another example, in some other implementations, one actual BV value of a previously coded block is stored as a first BV predictor candidate BV_{cand0}, and a default BV predictor is stored as a second BV predictor candidate BV_{cand1}. For example, the default BV

predictor is (BVx, 0), where BVx can be a negative offset equal to the width *W* of the current block or have some other non-zero value. After a new BV value has been used, the first BV predictor candidate BV_{cand0} is replaced with the BV value that was just used, but the second BV predictor candidate BV_{cand1} is unchanged. At the encoder side, for a current block, the encoder selects one of BV_{cand0} and BV_{cand1} to use as BV predictor. The encoder signals a flag value that indicates which one of BV_{cand0} and BV_{cand1} has been used by the encoder and should be used by the decoder. A decoder receives and parses the flag value and, based on the flag value, selects one of BV_{cand0} and BV_{cand1} to use as the BV predictor for the block. In this approach, the encoder and the decoder each store two BV predictor candidates BV_{cand0} and BV_{cand1}, and the first BV predictor candidate BV_{cand0} is updated during encoding or decoding.

[0158] In another approach, BV predictor candidates are assembled from among actual BV values of previously coded blocks that neighbor the current block in a picture. The encoder and decoder determine up to *x* actual BV values of neighboring blocks. The neighboring blocks are previously coded intra-BC-predicted blocks in the picture to the left of the current block, above the current block, to the above-right of the current block, and so on. The value of *x* depends on implementation and can be, for example, 2, 3, 4, 5 or some other number. If less than *x* neighboring blocks with actual BV values are available, the set of BV predictor candidates can include less than *x* BV predictor candidates.

[0159] Figure 15 shows an example of this approach. In Figure 15, three BV values (1531, 1541, 1551) of three neighboring intra-BC-predicted blocks (1530, 1540, 1550) provide the BV predictor candidates for merge mode for the BV predictor of a current block (1560) of a current frame (1510). If one of the neighboring blocks is unavailable (*e.g.*, not an intra-BC-predicted block, or outside the picture, tile or slice), in some cases another neighboring block can be added (*e.g.*, block to the above-left of the current block). As shown in Figure 15, the set of BV predictor candidates can include one or more default BV predictors (such as a default BV predictor (BVx, 0) with horizontal displacement and a default BV predictor (0, BVy) with vertical displacement). In Figure 15, index values idx 0 … idx 4 are assigned to the BV predictor candidates with BV values (1541, 1531, 1551) and default BV predictors according to expected likelihood of selection.

[0160] The treatment of redundant values among the BV predictor candidates depends on implementation. In some implementations, a set of BV predictor candidates for merge mode can include redundant values – that is, redundant BV predictor candidates are not removed or replaced. In other implementations, when assembling a set of BV predictor candidates for merge mode, an encoder and decoder do not add a redundant BV predictor value to the set. This approach can result in a smaller set of BV predictor candidates, which can reduce bitrate for coded index values. In still other implementations, when assembling a set of BV predictor candidates for merge mode, an encoder and decoder replace a BV predictor candidate that would be redundant to a BV predictor candidate already in the set with a different BV predictor candidate (*e.g.*, from another previously coded intra-BC-predicted block, from another neighboring intra-BC-predicted block, from another default BV predictor option). For example, if the BV value of the 3rd previous intra-BC-predicted block is the same as the BV value of the 1st previous intra-BC-predicted block or 2nd previous intra-BC-predicted block, the encoder/decoder instead adds the BV value of the 4th previous intra-BC-predicted block to the set of BV predictor candidates. If that BV value would be redundant, the encoder/decoder instead adds the BV value of the 5th, 6th, etc. previous intra-BC-predicted block. Or, the encoder/decoder adds another default BV predictor.

2. Example Encoding with Selected BV Predictors.

[0161] Figure 16a shows a generalized technique (1600) for encoding that includes merge mode for BV predictors. An image encoder or video encoder such as one described with reference to Figure 3 or Figures 5a-5b can perform the technique (1600).

[0162] To start, the encoder determines (1610) a set of multiple BV predictor candidates for a current intra-BC-predicted block of a picture. The current block is, for example, a part of a CU of a CTU. The encoder can determine the set of BV predictor candidates using one of the approaches described with reference to Figures 14 and 15, or using another approach. In general, the BV predictor candidates can include up to *x* actual BV values of previous blocks, respectively, where *x* depends on implementation (e.g., *x* is 2, 3, 4, 5, or some other number). The *x* previous blocks can be identified based at least in part on location in a neighborhood around the current block (*see, e.g.*, Figure 15) and/or decoding order compared to the current block (*see, e.g.*, Figure 14). The BV predictor candidates can also include one

or more default BV predictors (*e.g.*, each having a BV predictor component with a non-zero value).

[0163] The encoder selects (1620) one of the multiple BV predictor candidates to use for the current intra-BC-predicted block. For example, when the current block will be encoded using intra BC prediction with the selected BV predictor candidate, the encoder selects the BV predictor candidate for which the referenced intra-prediction region most closely matches the current block by some metric (*e.g.*, sum of absolute difference, mean squared error). Or, when the current block has a BV value (identified through BV estimation) that will be used for intra BC prediction, the encoder selects the BV predictor candidate that most closely matches the BV value for the current block. This selection results in the smallest BV differential, which tends to improve the efficiency of entropy coding.

[0164] The encoder encodes (1630) the current intra-BC-predicted block using the selected BV predictor candidate. Figure 16b shows details of the encoding (1630) of the current block in example implementations. The encoder checks (1632) whether the current block is encoded in skip mode. If so, the encoder encodes (1633) the current block using intra BC prediction with the selected BV predictor but not using any residual data. Otherwise (not skip mode), the encoder checks (1634) whether the current block is encoded in merge mode. If so, the encoder encodes (1635) the current block using intra BC prediction with the selected BV predictor. In merge mode, the encoder optionally encodes residual data for the current block. Otherwise (not merge mode), the encoder encodes (1637) the current block using intra BC prediction with the BV value for the current block, optionally encoding residual data for the current block.

[0165] The encoder outputs (1640) in a bitstream an index value that indicates the selected BV predictor candidate. When the set of BV predictor candidates includes two BV predictor candidates, the index value can be a flag value. Or, when there are more than two BV predictor candidates, the index value can be an integer value. The index value can be entropy coded or signaled as a fixed-length value. The index value can be (conditionally) signaled in the bitstream as a separate syntax element if an intra BC prediction mode syntax element indicates the current block is encoded using intra BC prediction mode. Or, the index

value can be signaled in the bitstream jointly with another syntax element (*e.g.*, a syntax element that indicates whether the current block is encoded using intra BC prediction mode).

[0166] The encoder also outputs any encoded data for the current block. A value in the bitstream can indicate whether the current block is encoded in skip mode. If the current block is a skip-mode block, the bitstream lacks a BV differential for the current block and lacks residual data for the current block. If the current block is a merge-mode block, the bitstream lacks a BV differential for the current block but may include residual data for the current block. Otherwise (current block is a non-skip-mode, non-merge-mode block), the bitstream includes a BV differential for the current block (indicating a difference between the selected BV predictor candidate and BV value for the current block) and may also include residual data for the current block.

[0167] When the encoder uses a data structure to track the actual BV values used for previous intra-BC-predicted blocks, the encoder can update the data structure with the BV value for the current block.

3. Example Decoding with Selected BV Predictors.

[0168] Figure 17a shows a generalized technique (1700) for decoding that includes merge mode for BV predictors. An image decoder or video decoder such as one described with reference to Figure 4 or Figure 6 can perform the technique (1700).

[0169] To start, the decoder receives (1710) from a bitstream an index value for a current block of a picture. The current block is, for example, a part of a CU of a CTU. The index value indicates a selected BV predictor candidate from among a set of multiple BV predictor candidates. When the set of BV predictor candidates includes two BV predictor candidates, the index value can be a flag value. Or, when there are more than two BV predictor candidates, the index value can be an integer value. The index value can be entropy coded, in which case the decoder entropy decodes the index value, or signaled as a fixed-length value. The index value can be (conditionally) signaled in the bitstream as a separate syntax element if an intra BC prediction mode syntax element indicates the current block has been encoded using intra BC prediction mode. Or, the index value can be signaled in the bitstream jointly

with another syntax element (*e.g.*, a syntax element that indicates whether the current block is encoded using intra BC prediction mode).

[0170] The decoder also receives any encoded data for the current block. The decoder entropy decodes the encoded data at the appropriate stage of decoding. A value in the bitstream can indicate whether the current block has been encoded in skip mode. If the current block is a skip-mode block, the bitstream lacks a BV differential for the current block and lacks residual data for the current block. If the current block is a merge-mode block, the bitstream lacks a BV differential for the current block but may include residual data for the current block. Otherwise (current block is a non-skip-mode, non-merge-mode block), the bitstream includes a BV differential for the current block and may also include residual data for the current block.

[0171] The decoder determines (1720) the set of multiple BV predictor candidates for the current block. The decoder can determine the set of BV predictor candidates using one of the approaches described with reference to Figures 14 and 15, or using another approach. In general, the BV predictor candidates can include up to x actual BV values of previous blocks, where x depends on implementation (e.g., x is 2, 3, 4, 5, or some other number). The x previous blocks can be identified based at least in part on location in a neighborhood around the current block (*see*, *e.g.*, Figure 15) and/or decoding order compared to the current block (*see*, *e.g.*, Figure 14). The BV predictor candidates can also include one or more default BV predictors (*e.g.*, each having a BV predictor component with a non-zero value).

[0172] Based on the index value, the decoder selects (1730) one of the multiple BV predictor candidates to use for the current block. The decoder decodes (1740) the current block using the selected BV predictor candidate. Figure 17b shows details of the decoding (1740) of the current block in example implementations. The decoder checks (1742) whether the current block has been encoded in skip mode. If so, the decoder decodes (1743) the current block using intra BC prediction with the selected BV predictor but not using any residual data. Otherwise (not skip mode), the decoder checks (1744) whether the current block has been encoded in merge mode. If so, the decoder decodes (1745) the current block using intra BC prediction with the selected BV predictor. In merge mode, the decoder decodes residual data (if present) for the current block. Otherwise (not merge mode), the

decoder decodes (1747) the current block using intra BC prediction with the BV value for the current block (decoded with a BV differential relative to the selected BV predictor), also decoding residual data (if present) for the current block.

[0173] When the decoder uses a data structure to track the actual BV values used for previous intra-BC-predicted blocks, the decoder can update the data structure with the BV value for the current block.

4. Example Implementations of Merge Mode.

[0174] In HEVC implementations, the syntax and semantics for CUs can be modified to signal syntax elements for merge mode for BV prediction and/or skip mode for intra-BC-predicted blocks.

[0175] In the draft HEVC specification in JCTVC-O1005_v3, cu_skip_flag[x0][y0] indicates whether a current CU is skipped, but the flag is signaled only if the current slice is a predictive ("P") slice or bi-predictive ("B") slice. The array indices x0, y0 specify the location (x0, y0) of the top-left luma sample of the considered CB relative to the top-left luma sample of the picture. If cu_skip_flag[x0][y0] is 1, the current CU is skipped, and no more syntax elements are parsed after cu_skip_flag[x0][y0] for the CU, except for the merging candidate index merge_idx[x0][y0]. If cu_skip_flag[x0][y0] is 0, the CU is not skipped. When cu_skip_flag[x0][y0] is not present, it is inferred to be 0.

[0176] According to one possible modification, cu_skip_flag[x0][y0] is also signaled if the current slice is an intra ("I") slice. When intra BC prediction is enabled (according to the flag intra_block_copy_enabled_flag in the sequence parameter set), if the value of cu_skip_flag[x0][y0] is 1, the current CU is skipped, and no more syntax elements are parsed after cu_skip_flag[x0][y0] for the CU, except the BV predictor index value idx. In particular, no BV difference is signaled for the current, skipped CU. On the other hand, if the value of cu_skip_flag[x0][y0] is 0, the current CU is not skipped.

[0177] In the draft HEVC specification in JCTVC-O1005_v3, intra_bc_flag[x0][y0] indicates whether a current CU is coded in intra BC prediction mode. The array indices x0, y0 specify the location (x0, y0) of the top-left luma sample of the considered CB relative to

the top-left luma sample of the picture. If intra_bc_flag[x0] y0] is 1, the current CU is coded in intra BC prediction mode. If intra_bc_flag[x0][y0] is 0, the current CU is coded with intra spatial prediction. When not present, the value of intra_bc_flag is inferred to be 0.

[0178] According to one possible modification for intra_bc_flag[x0][y0], when intra_bc_flag[x0][y0]is 1, syntax for the current CU includes a BV predictor index value idx that indicates a selection of BV predictor candidate. The index value idx can be a flag value, to indicate a selection between two BV predictor candidates, or an integer value, to indicate a selection among more than two BV predictor candidates. The index value can be entropy coded using, for example, arithmetic coding.

[0179] Or, according to another possible modification for intra_bc_flag[x0][y0], the syntax element is an integer value instead of a binary flag. If intra_bc_flag[x0][y0] is 0, the current CU is coded with intra spatial prediction, as in JCTVC-O1005_v3. On the other hand, if intra_bc_flag[x0][y0] is greater than 0, then the current CU is coded in intra BC prediction mode, and intra_bc_flag[x0][y0] also indicates the BV predictor index value idx. For example, if intra_bc_flag[x0][y0] is 1, then idx is 1; if intra_bc_flag[x0][y0] is 2, then idx is 2; and so on. In other words, the BV predictor index value idx is jointly coded with the binary value for spatial or BC prediction in a single syntax element.

[0180] For either way of modifying intra_bc_flag[x0][y0] and signaling the index value idx, the variable BvIntraPredictor[idx][compIdx] specifies the BV predictor candidate to be used for intra BC prediction of the current CU. For example, in the example of Figure 14, BvIntraPredictor[0][compIdx], BvIntraPredictor[1][compIdx] and BvIntraPredictor[2][compIdx] store the BV values of up to three of the most recent previous intra-BC-predicted blocks. Or, in the example of Figure 15, BvIntraPredictor[0][compIdx], BvIntraPredictor[1][compIdx] and BvIntraPredictor[2][compIdx] store the BV values of up to three neighboring intra-BC-predicted blocks. BvIntraPredictor[3][compIdx] and BvIntraPredictor[4][compIdx] store the default BV predictors in the horizontal and vertical directions, respectively. The horizontal BV component is assigned compIdx = 0, and the vertical BV component is assigned compIdx = 1.

[0181] For a skip-mode block or merge-mode block, no BV differential is signaled for the block. The BV value for the block is the selected BV predictor candidate.

[0182] For a non-skip-mode, non-merge-mode block, the BV differential is signaled in the bitstream. The variable BvdIntra[x0][y0][compIdx] specifies the BV differential. The array indices x0, y0 specify the location (x0, y0) of the top-left luma sample of the considered PB relative to the top-left luma sample of the picture. The horizontal BV differential component is assigned compIdx = 0, and the vertical BV differential component is assigned compIdx = 1. The BV value for the block is BvIntra[x0][y0][compIdx] = BvdIntra[x0][y0][compIdx] + BvIntraPredictor[idx][compIdx], for compIdx = 0..1.

[0183] The variable BvIntraPredictor[idx][compIdx] can then be updated to include the BV value for the current CU.

5. Alternatives and Variations.

[0184] When a current block has a BV value that is identified through BV estimation, the encoder can use any form of BV estimation to identify the BV value for the current block. During the BV estimation for the current block, the encoder can start by evaluating one or more BV predictor candidates that are available for the current block. Starting BV estimation with BV predictor candidate(s) can help quickly identify a suitable BV value, avoiding evaluation of other BV values. The encoder evaluates other BV values as needed. The encoder can use a data structure to track BV predictor candidates. When a BV value can be applied for different sizes of blocks (*e.g.*, for a 32x32 CU, 16x16 CU, 8x8 CU or smaller TU or CU), the data structure can include different BV predictor candidates for the different sizes of blocks (*e.g.*, one or more BV predictor candidates for a 32x32 CU, one or more BV predictor candidates for a 16x16 CU, and so on). For the current block, the encoder first evaluates the BV predictor candidate(s) stored in the data structure for the level (size) of the current block, then proceeds with evaluation of other BV values if needed.

[0185] In view of the many possible embodiments to which the principles of the disclosed invention may be applied, it should be recognized that the illustrated embodiments

are only preferred examples of the invention and should not be taken as limiting the scope of the invention. Rather, the scope of the invention is defined by the following claims. We therefore claim as our invention all that comes within the scope and spirit of these claims.

CLAIMS

We claim:

1. In a computing device with a video encoder or image encoder, a method comprising:

determining a default block vector ("BV") predictor for a current block of a picture, the default BV predictor including a BV predictor component with a non-zero value; and

encoding the current block using the default BV predictor.

2. The method of claim 1 wherein the encoding the current block using the default BV predictor includes:

performing intra block copy ("BC") prediction using a BV value for the current block;

determining a BV differential for the current block using the BV value for the current block and the default BV predictor for the current block; and

encoding the BV differential for the current block.

3. The method of claim 1 wherein the encoding the current block using the default BV predictor includes performing intra BC prediction using the default BV predictor.

4. In a computing device with a video decoder or image decoder, a method comprising:

determining a default block vector ("BV") predictor for a current block of a picture, the default BV predictor including a BV predictor component with a non-zero value; and

decoding the current block using the default BV predictor.

5. The method of claim 4 wherein the decoding the current block using the default BV predictor includes:

decoding a BV differential for the current block;

combining the decoded BV differential for the current block with the default BV predictor for the current block to reconstruct a BV value for the current block; and

performing intra block copy ("BC") prediction using the BV value for the current block.

6. The method of claim 4 wherein the decoding the current block using the default BV predictor includes performing intra BC prediction using the default BV predictor.

7. The method of any of claims 1-6 wherein the default BV predictor is part of a set of multiple BV predictor candidates assembled.

8. The method of any of claims 1-6 further comprising:

checking whether an actual BV value for a previous block of the picture is available, wherein the default BV predictor for the current block is used only if no actual BV value for any previous block of the picture is available.

9. The method of claim 8 wherein the actual BV value for the previous block is available if (1) the previous block and the current block are part of a given coding tree unit, and (2) prediction mode of the previous block is intra BC prediction mode.

10. The method of any of claims 1-9 wherein the current block is part of a coding unit.

11. The method of any of claims 1-10 wherein the BV predictor component with the non-zero value is a horizontal BV component.

12. The method of any of claims 1-10 wherein the BV predictor component with the non-zero value is a vertical BV component.

13. The method of any of claims 1-10 wherein the non-zero value is height or width of the current block.

14. The method of any of claims 1-10 wherein the non-zero is a fixed value independent of size of the current block.

15. In a computing device with a video encoder or image encoder, a method comprising:

using block vector ("BV") prediction to determine a BV value for a current block of a picture, the BV value for the current block indicating a displacement to a region within the picture;

encoding the current block using intra block copy prediction with the BV value; and

outputting in a bitstream an indication that the current block is encoded in skip mode, wherein the bitstream lacks a BV differential and residual data for the current block.

16. In a computing device with a video decoder or image decoder, a method comprising:

receiving from a bitstream an indication that a current block of a picture is encoded in skip mode, wherein the bitstream lacks a block vector ("BV") differential and residual data for the current block;

using BV prediction to determine a BV value for the current block, the BV value for the current block indicating a displacement to a region within the picture; and

decoding the current block using intra block copy prediction with the BV value.

17. The method of claim 15 or 16 wherein the bitstream includes an index value that indicates a selection of a BV predictor candidate, from a set of multiple BV predictor candidates, to use as the BV value for the current block.

18. The method of claim 17 wherein the index value is a flag value, the set of multiple BV predictor candidates having two BV predictor candidates.

19. The method of claim 17 wherein the index value is an integer value, the set of multiple BV predictor candidates having more than two BV predictor candidates.

20. The method of claim 19 wherein the multiple BV predictor candidates include one or more default BV predictors, each of the one or more default BV predictors including a BV predictor component with a non-zero value.

21. In a computing device with a video encoder or image encoder, a method comprising:

determining a set of multiple block vector ("BV") predictor candidates for a current block of a picture;

selecting one of the multiple BV predictor candidates to use for the current block;

encoding the current block using the selected BV predictor candidate; and

outputting in a bitstream an index value that indicates the selected BV predictor candidate.

22. In a computing device with a video decoder or image decoder, a method comprising:

receiving from a bitstream an index value;

determining a set of multiple block vector ("BV") predictor candidates for a current block of a picture;

based on the index value, selecting one of the multiple BV predictor candidates to use for the current block; and

decoding the current block using the selected BV predictor candidate.

23. The method of claim 21 or 22 wherein the current block is a merge-mode block, and wherein the bitstream lacks a BV differential for the current block.

24. The method of claim 23 wherein the bitstream lacks residual data for the current block.

25. The method of claim 23 wherein the bitstream includes residual data for the current block.

26. The method of claim 21 or 22 wherein a value in the bitstream indicates the current block is a skip-mode block, wherein the bitstream lacks a BV differential for the current block, and wherein the bitstream lacks residual data for the current block.

27. The method of claim 21 or 22 wherein the bitstream includes a BV differential for the current block, the BV differential indicating a difference between the selected BV predictor candidate and a BV value for the current block.

28. The method of any of claims 21-27 wherein the index value is a flag value, the set of multiple BV predictor candidates having two BV predictor candidates.

29. The method of any of claims 21-27 wherein the index value is an integer value, the set of multiple BV predictor candidates having more than two BV predictor candidates.

30. The method of any of claims 21-29 wherein the multiple BV predictor candidates include up to x actual BV values of previous blocks.

31. The method of claim 30 wherein x is at least 2.

32. The method of claim 30 wherein the previous blocks are identified based at least in part on location in a neighborhood around the current block.

33. The method of claim 30 wherein the previous blocks are identified based at least in part on decoding order compared to the current block.

34. The method of any of claims 21-33 wherein the multiple BV predictor candidates include one or more default BV predictors, each of the one or more default BV predictors including a BV predictor component with a non-zero value.

35. The method of any of claims 21-34 wherein the index value is signaled in the bitstream as part of a syntax element that indicates whether the current block is encoded using intra block copy prediction mode.

36. The method of any of claims 21-34 wherein the index value is conditionally signaled in the bitstream if an intra block copy prediction mode syntax element indicates the current block is encoded using intra block copy prediction mode.

37. The method of any of claims 21-36 further comprising:
updating a data structure used to track actual BV values of previous blocks.

38. A computing device adapted to perform the method of any one of claims 1-37.

39. One or more computer-readable media storing computer-executable instructions for causing a computing device programmed thereby to perform the method of any one of claims 1-37.

Figure 1

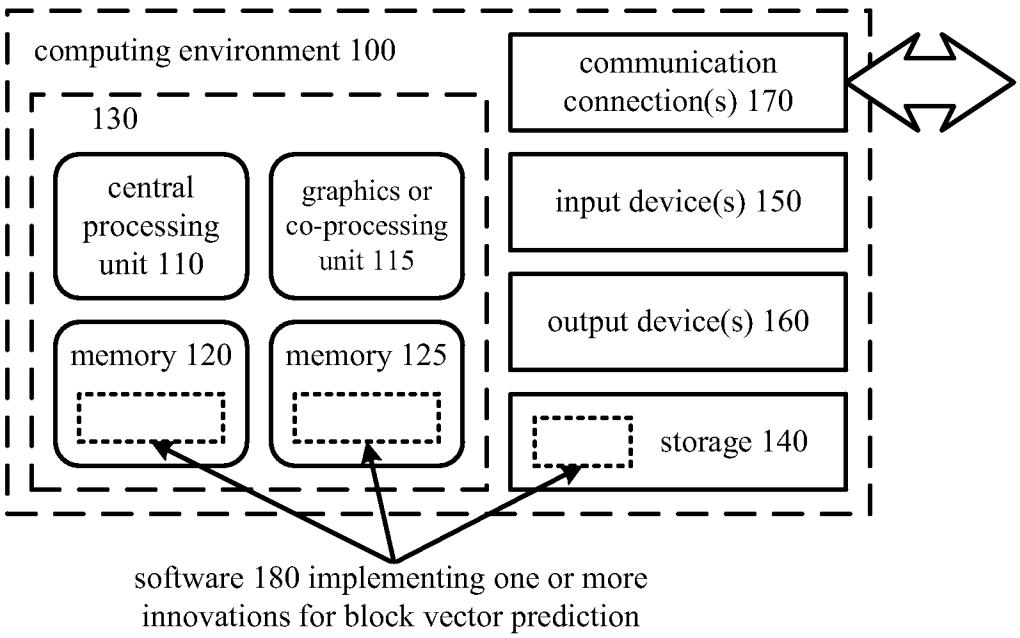


Figure 2a

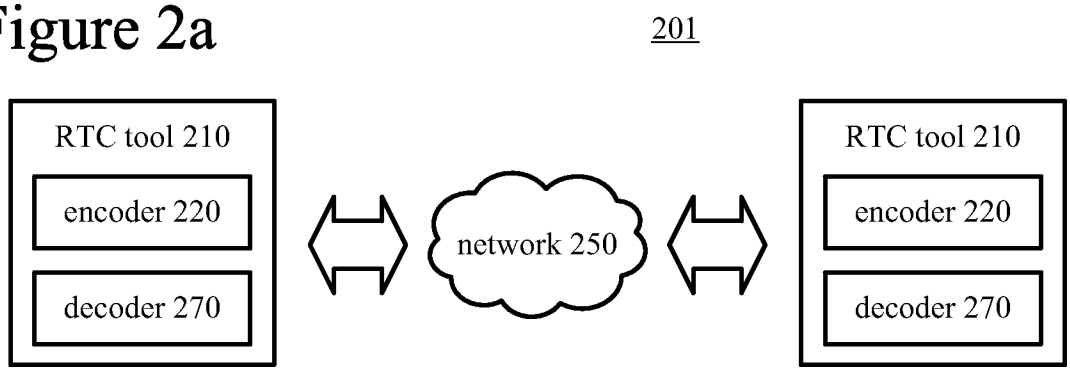


Figure 2b

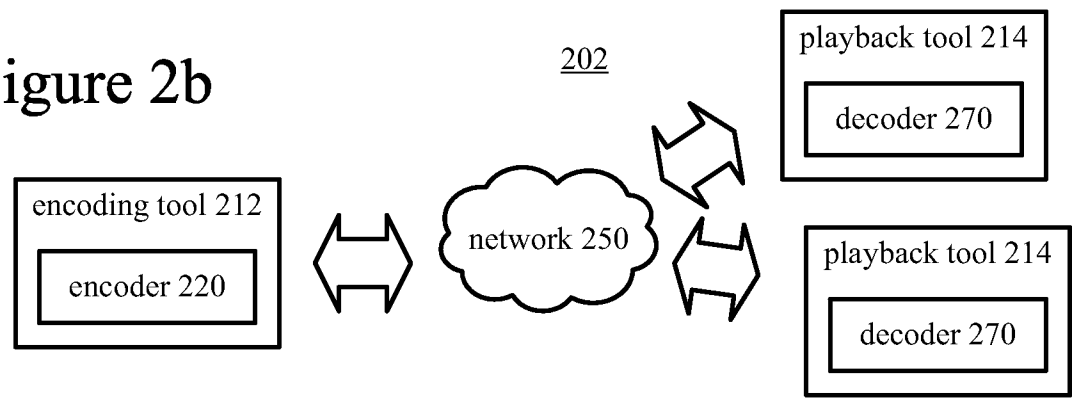


Figure 3

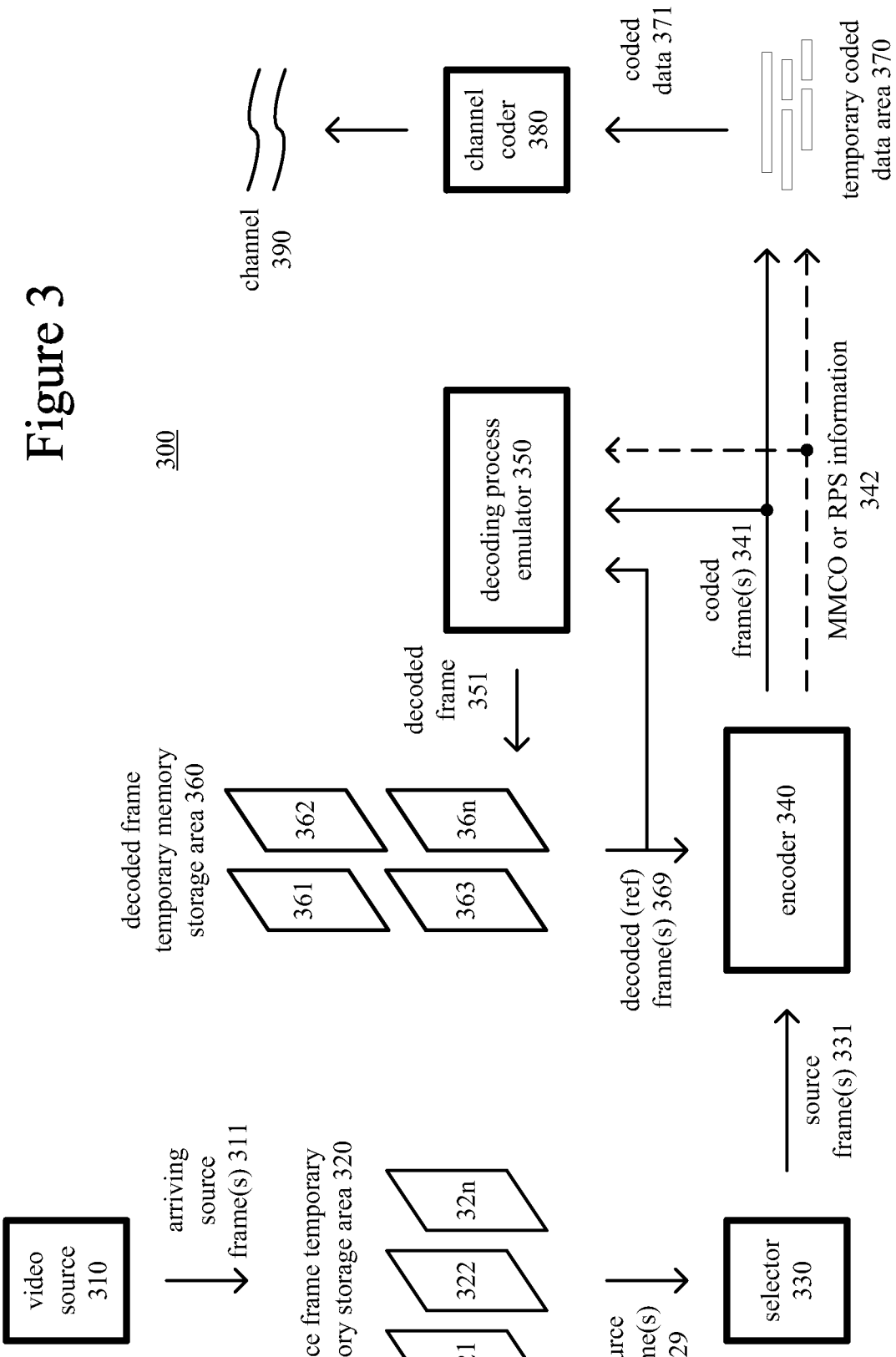

300
video source 310
arriving source frame(s) 311
source frame temporary memory storage area 320
321
322
32n
source frame(s) 329
selector 330
source frame(s) 331
encoder 340
decoded (ref) frame(s) 369
decoded frame temporary memory storage area 360
361
362
363
36n
decoded frame 351
decoding process emulator 350
coded frame(s) 341
MMCO or RPS information 342
temporary coded data area 370
coded data 371
channel coder 380
channel 390

Figure 4

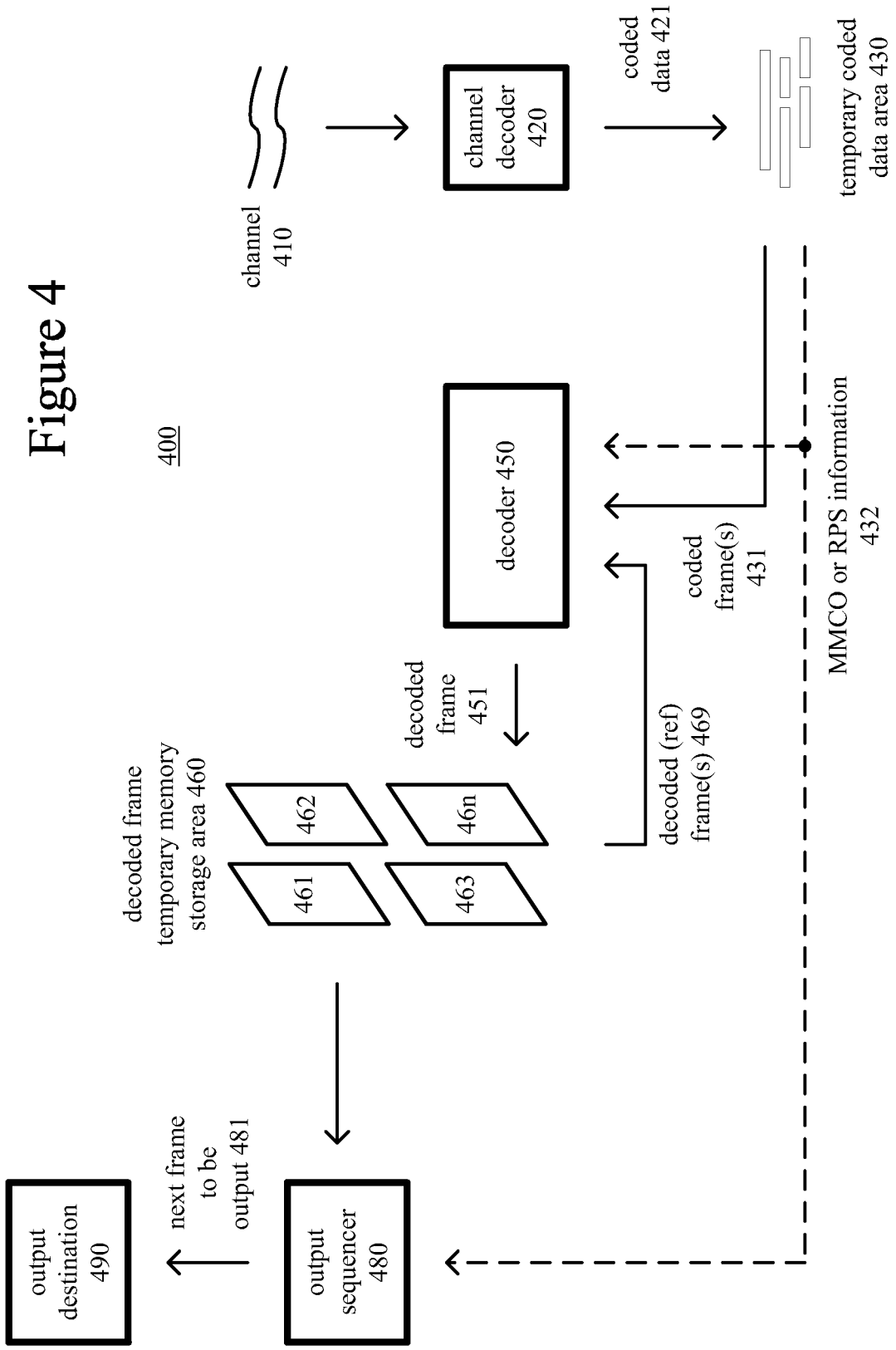

400
channel 410
channel decoder 420
coded data 421
temporary coded data area 430
decoder 450
decoded frame temporary memory storage area 460
461
462
463
46n
decoded frame 451
decoded (ref) frame(s) 469
coded frame(s) 431
MMCO or RPS information 432
output sequencer 480
next frame to be output 481
output destination 490

Figure 5a

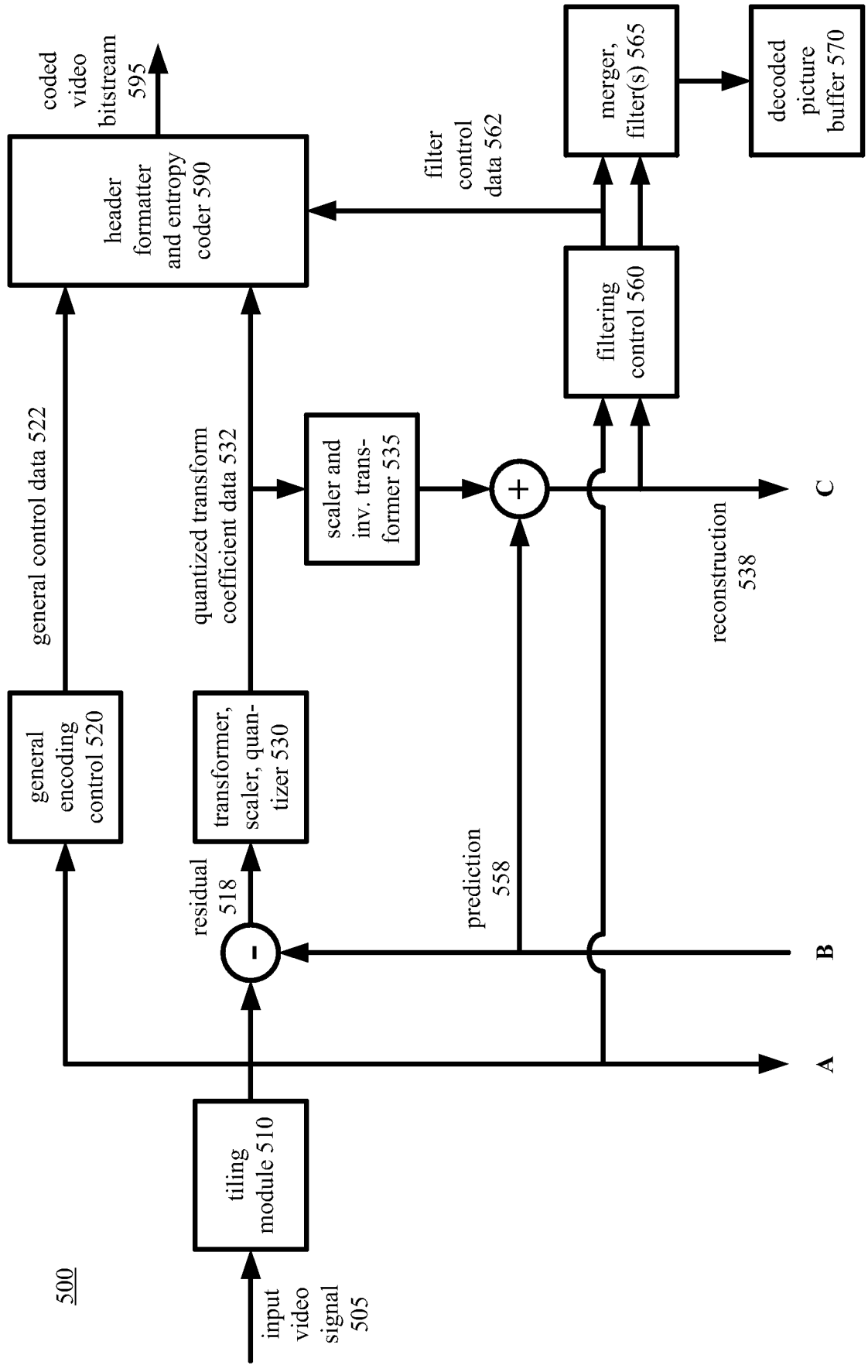
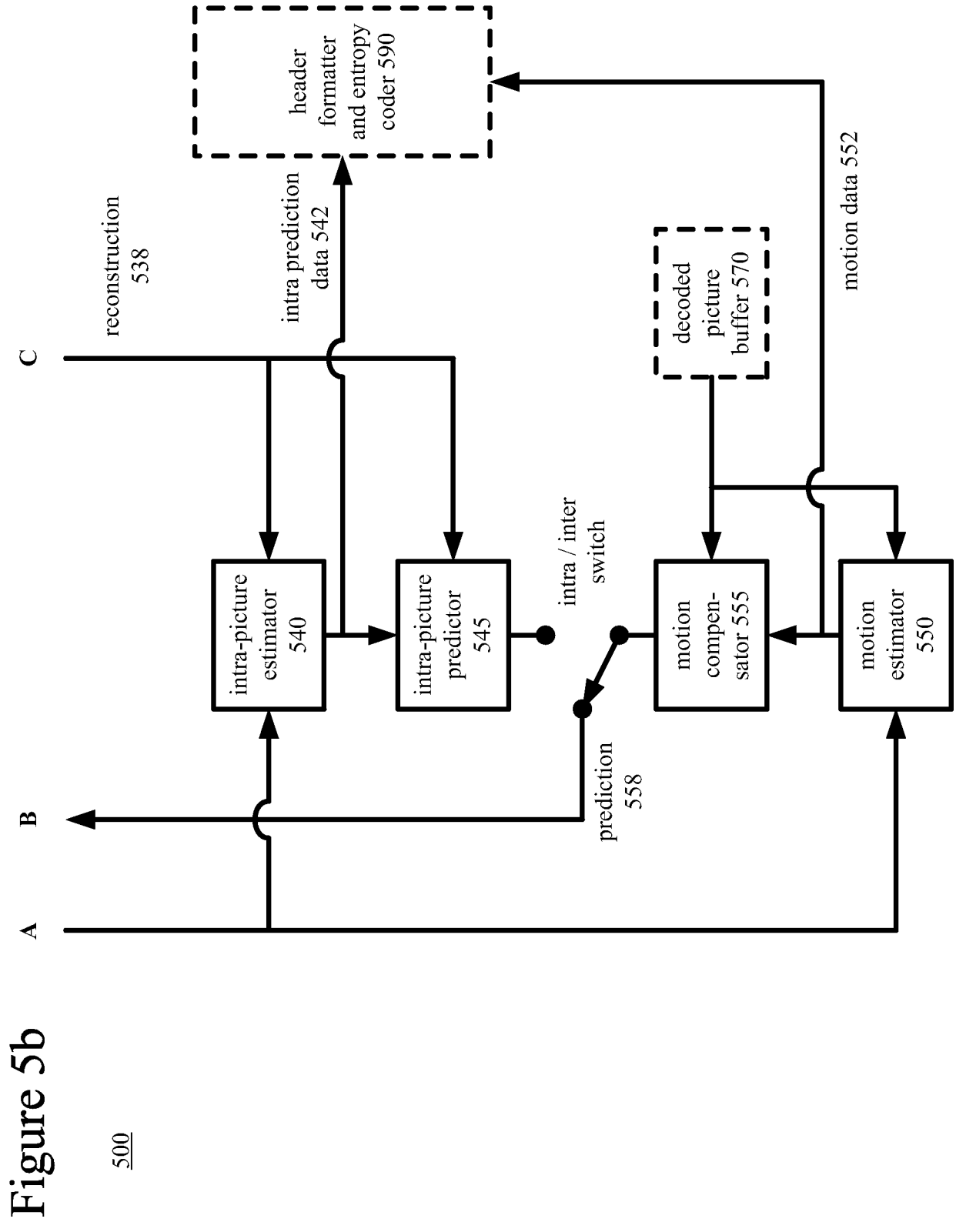

500
input video signal 505
tiling module 510
general encoding control 520
transformer, scaler, quantizer 530
scaler and inv. transformer 535
header formatter and entropy coder 590
coded video bitstream 595
filtering control 560
merger, filter(s) 565
decoded picture buffer 570
general control data 522
quantized transform coefficient data 532
filter control data 562
residual 518
prediction 558
reconstruction 538
A
B
C

Figure 5b


500
A
B
C
reconstruction 538
intra prediction data 542
header formatter and entropy coder 590
decoded picture buffer 570
motion data 552
intra-picture estimator 540
intra-picture predictor 545
intra / inter switch
motion compen-sator 555
motion estimator 550
prediction 558

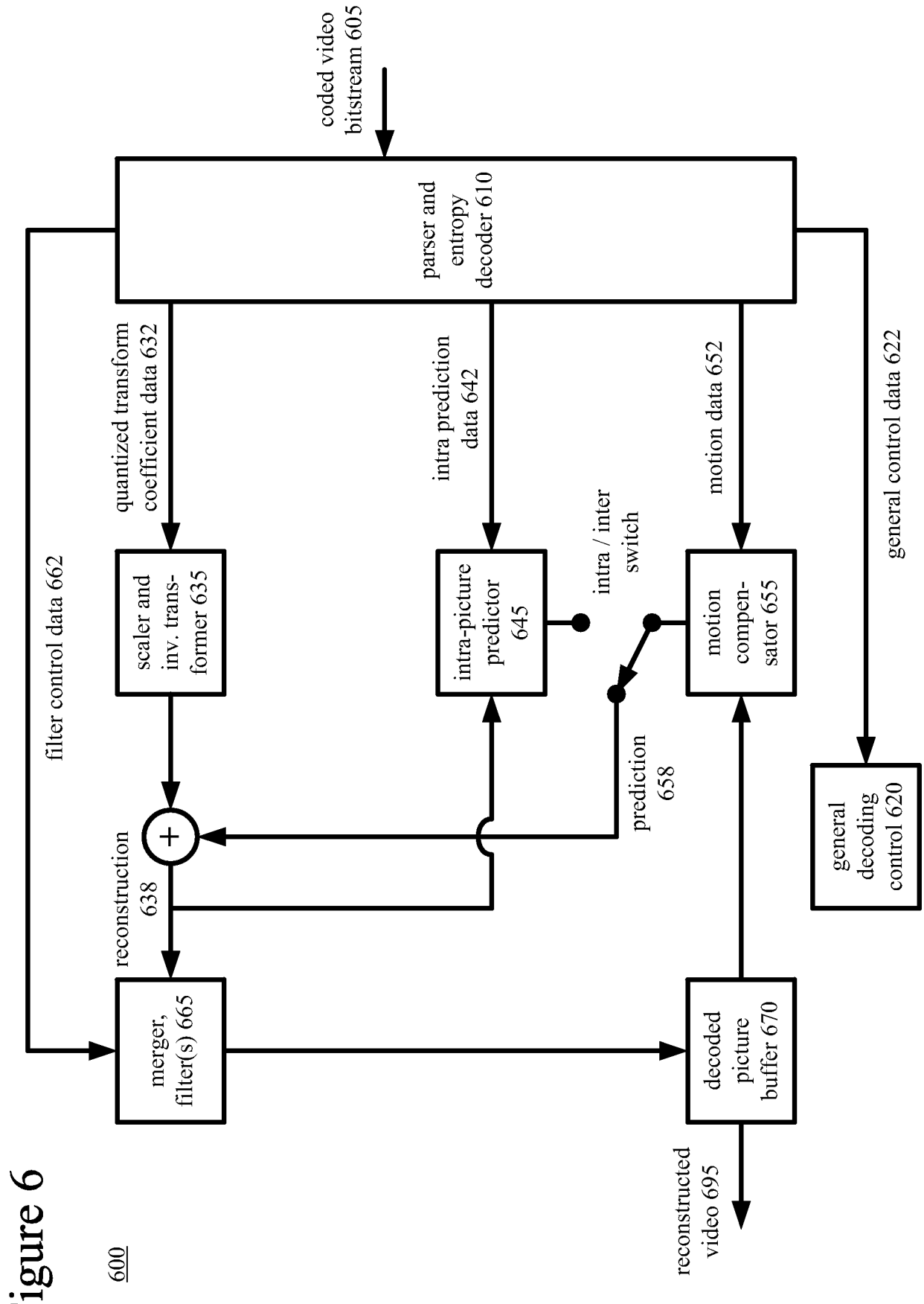
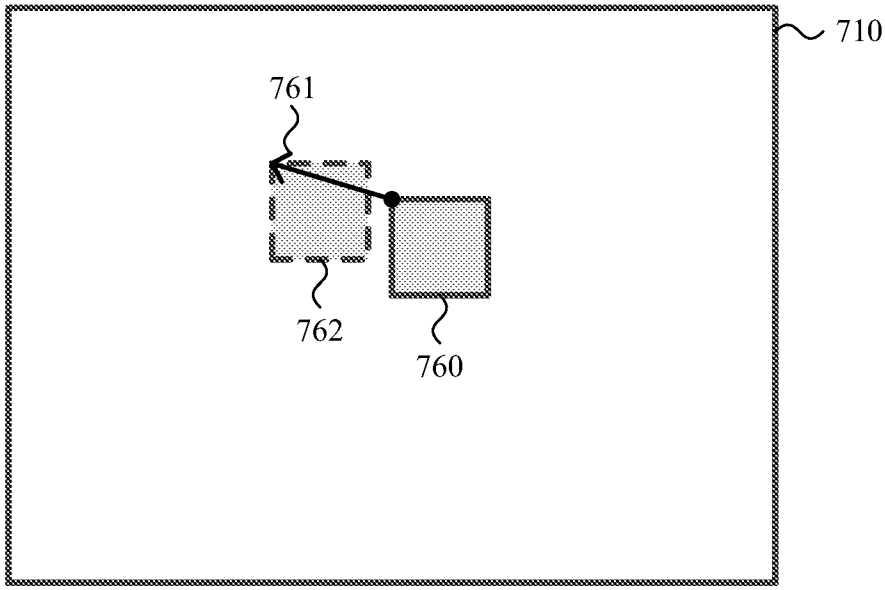

coded video bitstream 605
parser and entropy decoder 610
quantized transform coefficient data 632
intra prediction data 642
motion data 652
general control data 622
filter control data 662
scaler and inv. trans-former 635
intra-picture predictor 645
intra / inter switch
motion compen-sator 655
reconstruction 638
prediction 658
general decoding control 620
merger, filter(s) 665
decoded picture buffer 670
reconstructed video 695
600

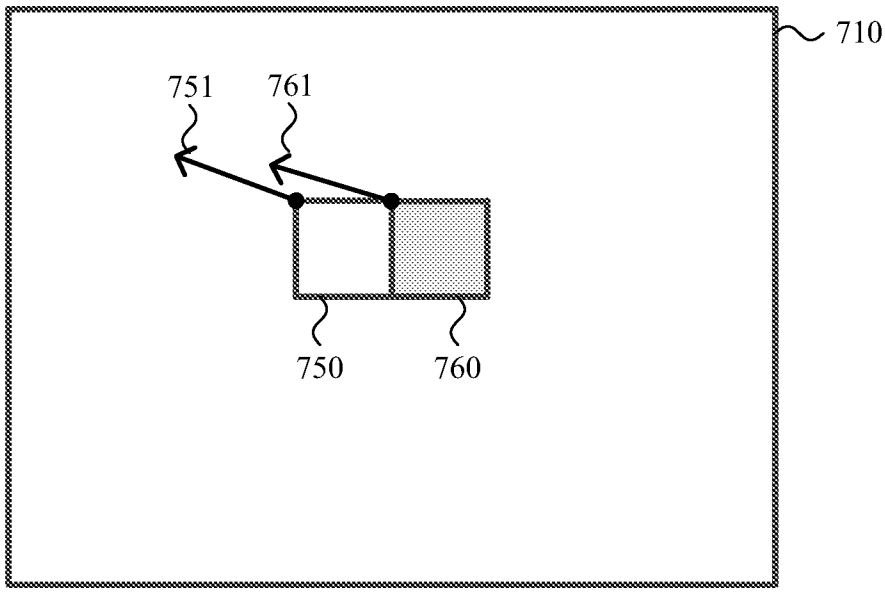
Figure 6

Figure 7a



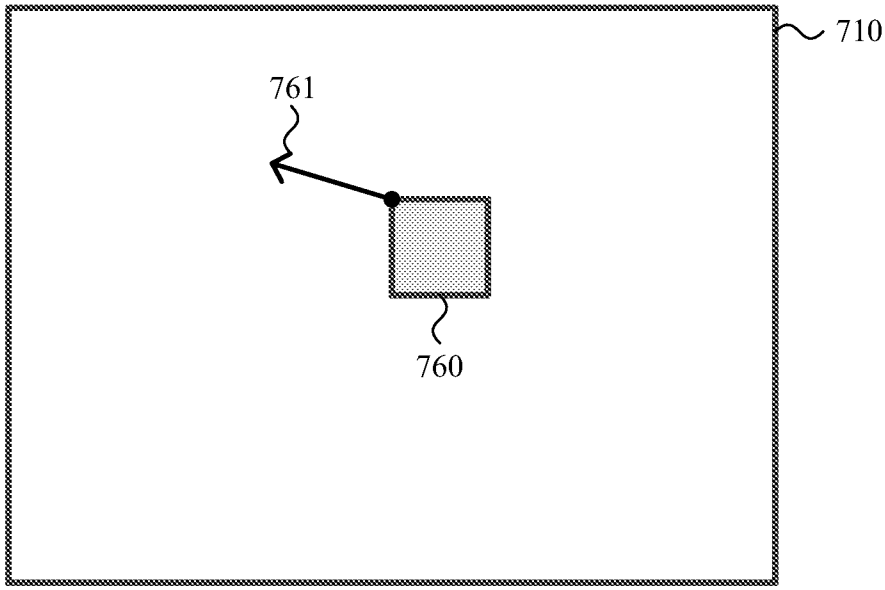
BV (761) for current block (760) of current frame (710), indicating a displacement to a matching block (762) in the current frame (710)

Figure 7b



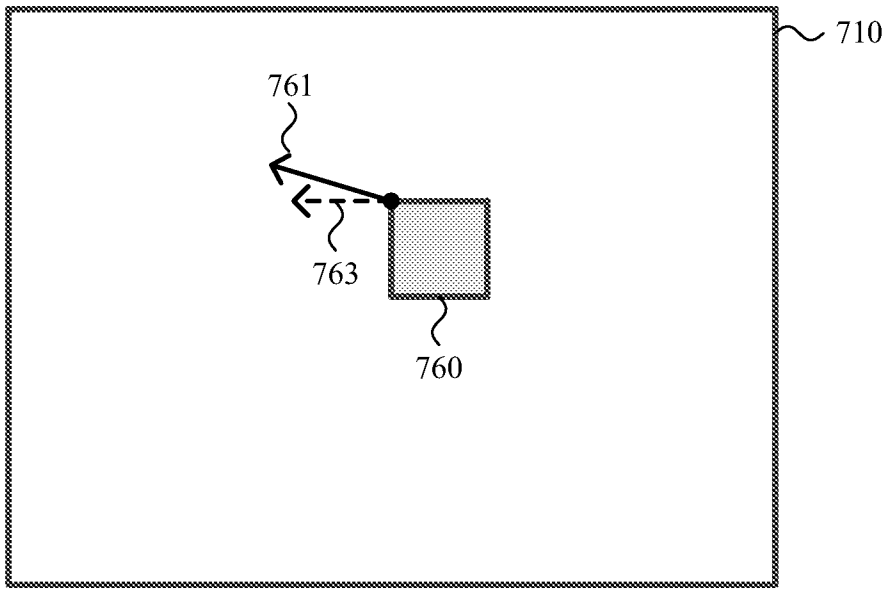
BV (751) of previous intra BC predicted block (750) used as BV predictor for BV (761) for current block (760) of current frame (710)

Figure 7c



default BV predictor of (0, 0) for BV (761) for current block (760) of current frame (710)

Figure 7d



default BV predictor (763) with non-zero horizontal component for BV (761) for current block (760) of current frame (710)

Figure 8 800

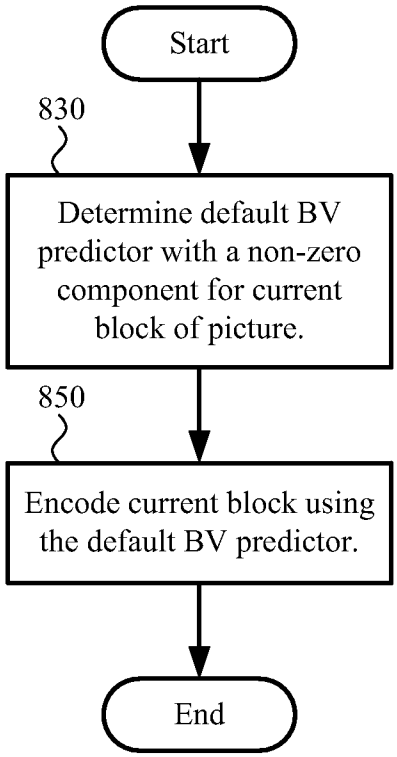


Figure 9 900

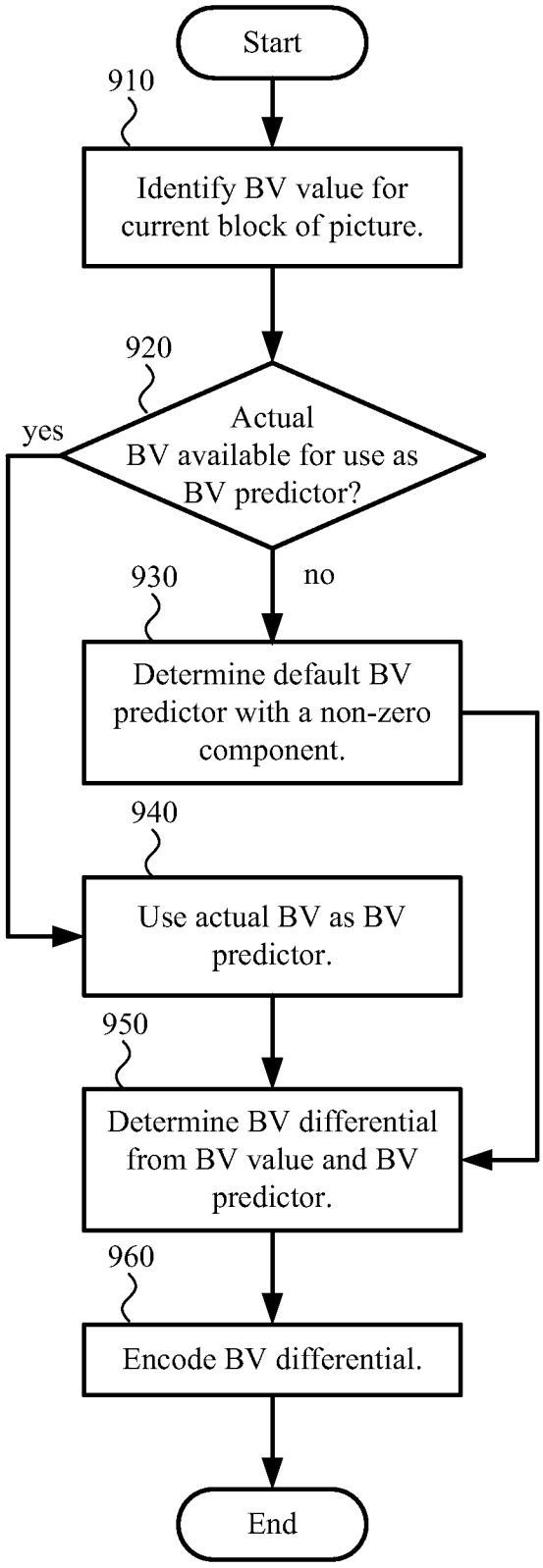


Figure 10 1000

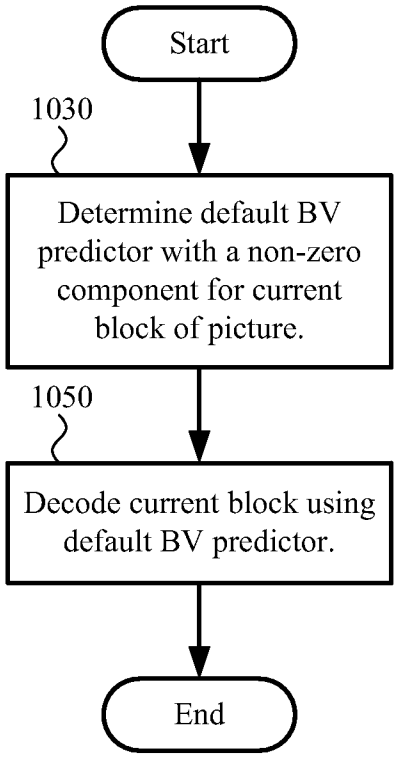


Figure 11 1100

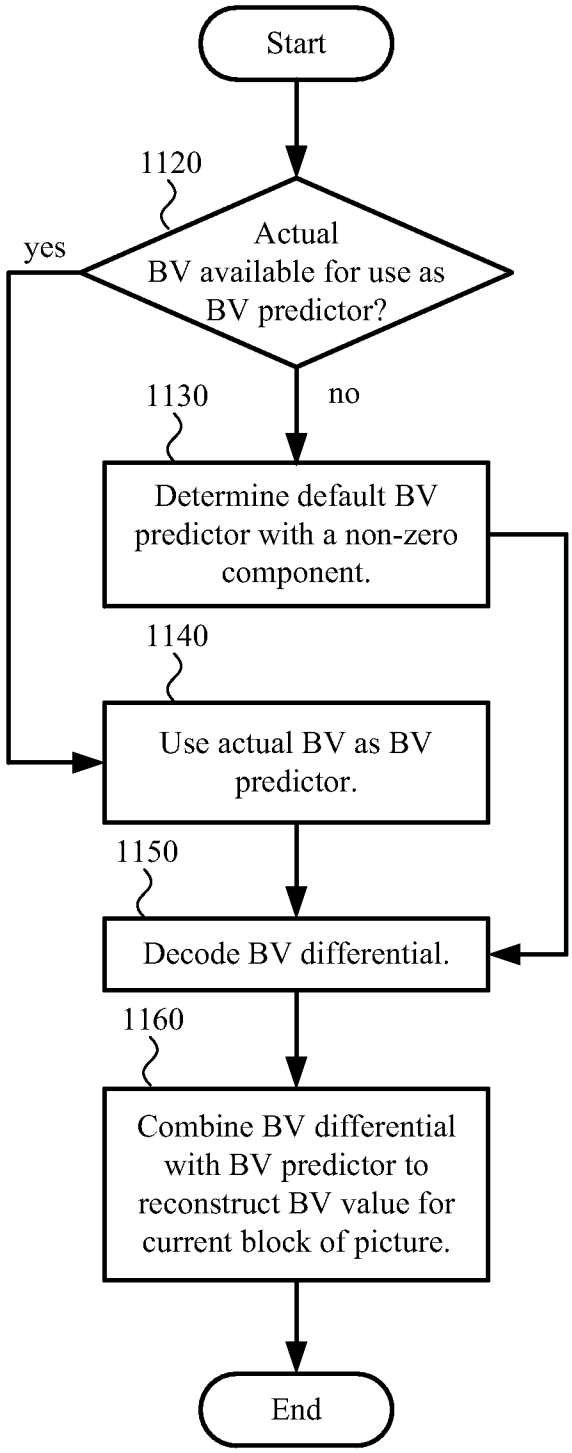


Figure 12 1200

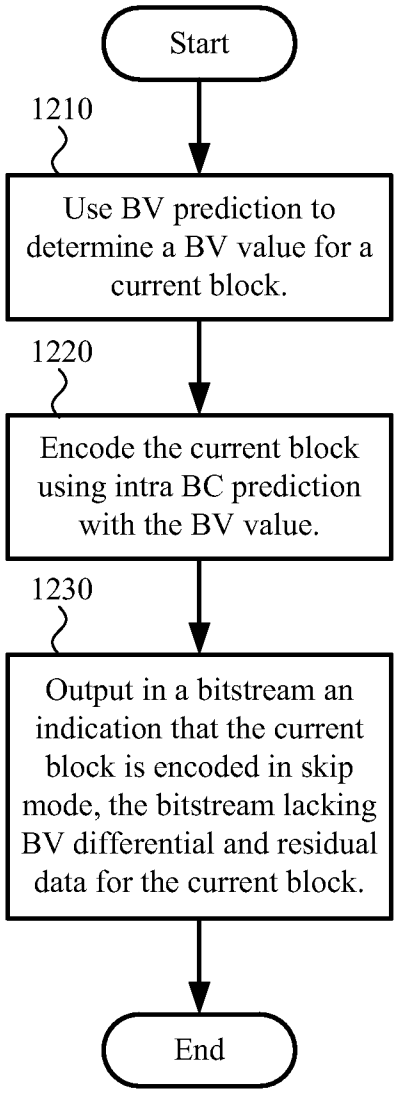


Figure 13 1300

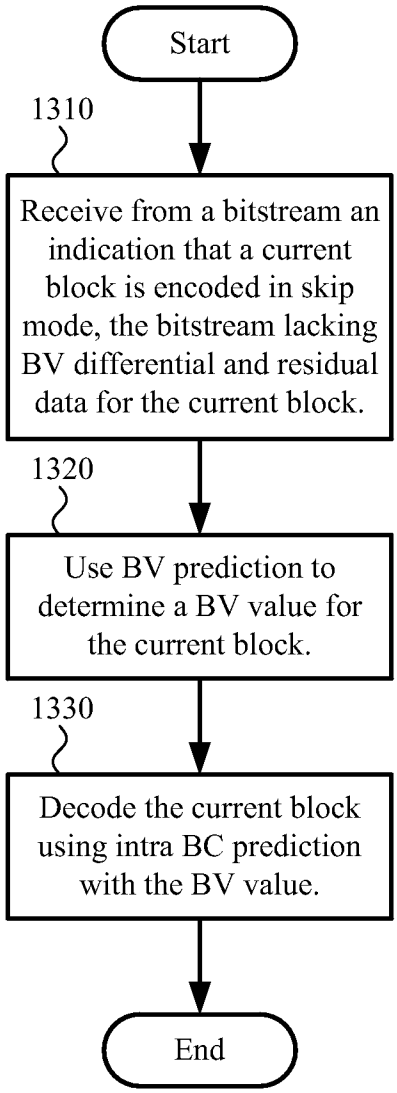
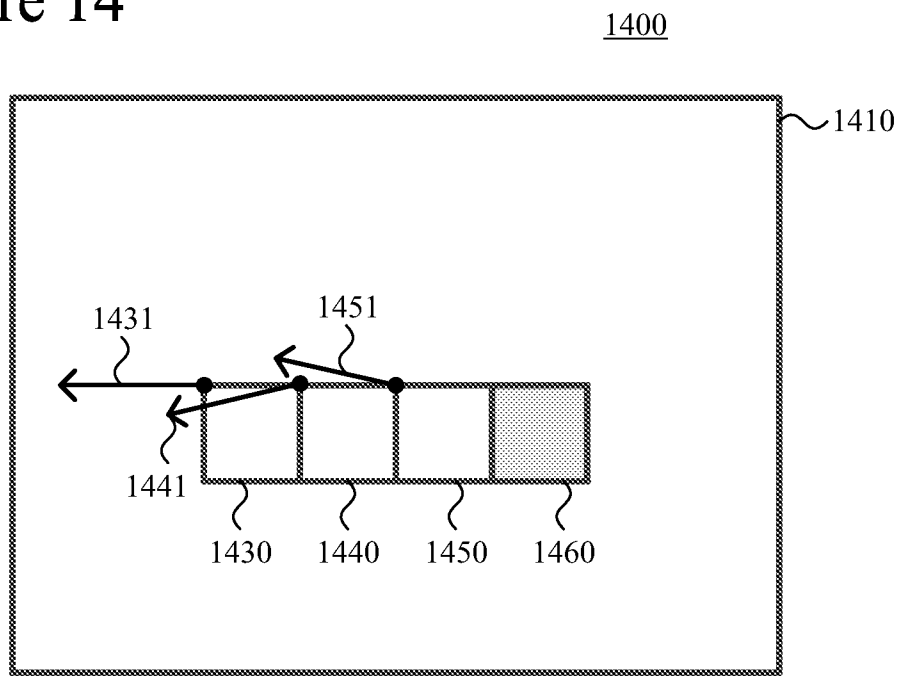


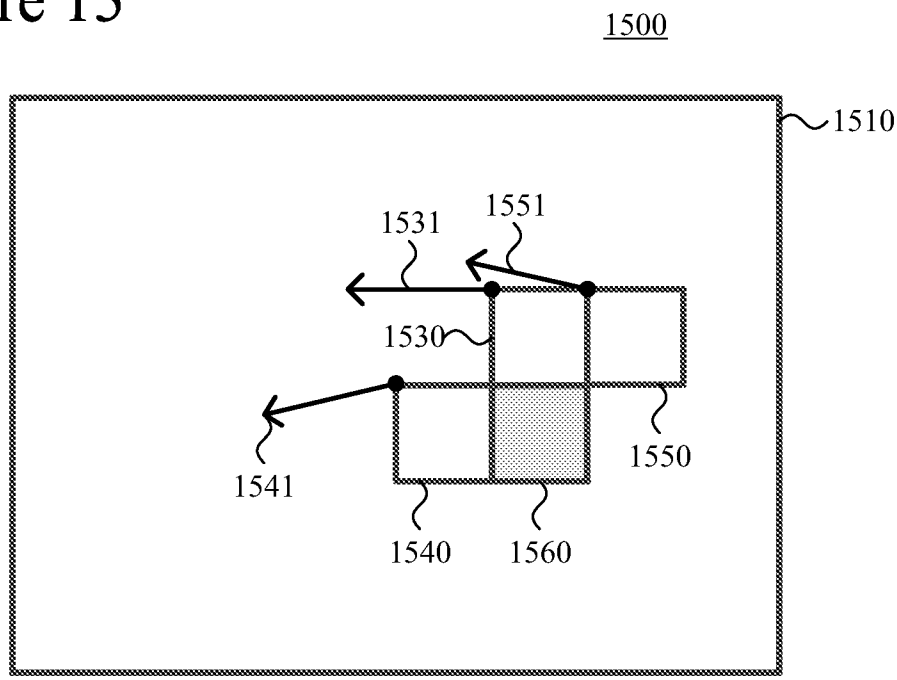
Figure 14



BV predictor candidates from BV values (1431, 1441, 1451) of three previous intra-BC-predicted blocks according to merge mode for BV for current block (1460)

merge mode index value:	block vector (BV) predictor
idx 0 :	BV (1451) from 1st previous intra-BC-predicted block (1450)
idx 1 :	BV (1441) from 2nd previous intra-BC-predicted block (1440)
idx 2 :	BV (1431) from 3rd previous intra-BC-predicted block (1430)
idx 3 :	default BV predictor with horizontal displacement
idx 4 :	default BV predictor with vertical displacement

Figure 15



BV predictor candidates from BV values (1531, 1541, 1551) of three neighboring intra-BC-predicted blocks according to merge mode for BV for current block (1560)

merge mode index value:	block vector (BV) predictor
idx 0 :	BV (1541) from 1st neighboring intra-BC-predicted block (1540)
idx 1 :	BV (1531) from 2nd neighboring intra-BC-predicted block (1530)
idx 2 :	BV (1551) from 3rd neighboring intra-BC-predicted block (1550)
idx 3 :	default BV predictor with horizontal displacement
idx 4 :	default BV predictor with vertical displacement

Figure 16a 1600

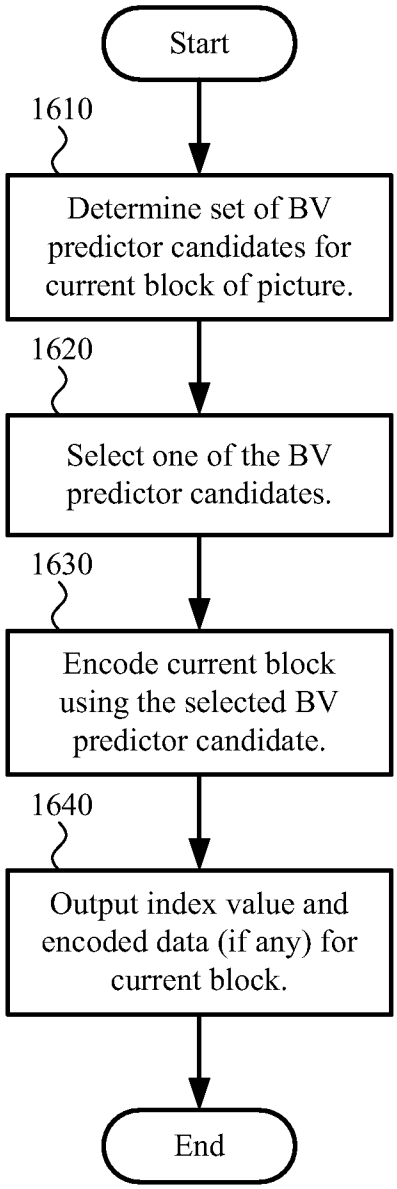


Figure 17a 1700

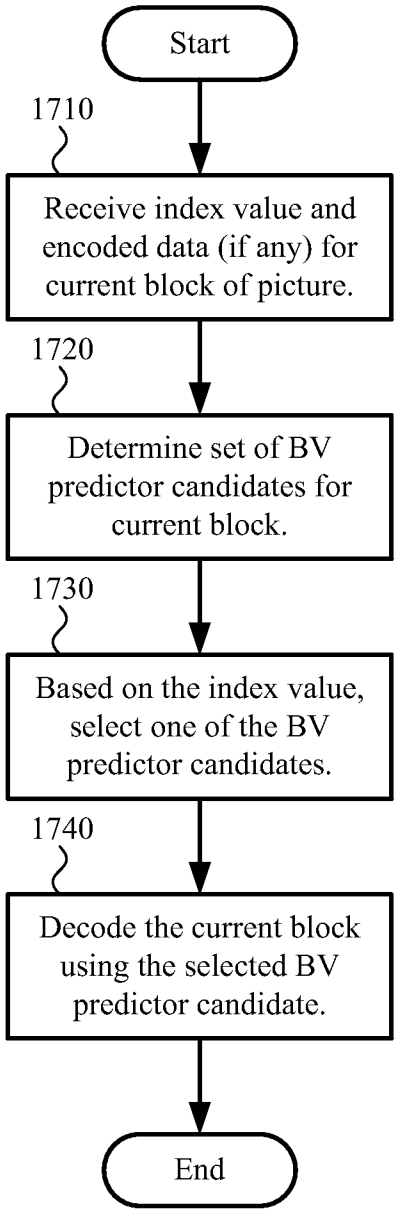


Figure 16b

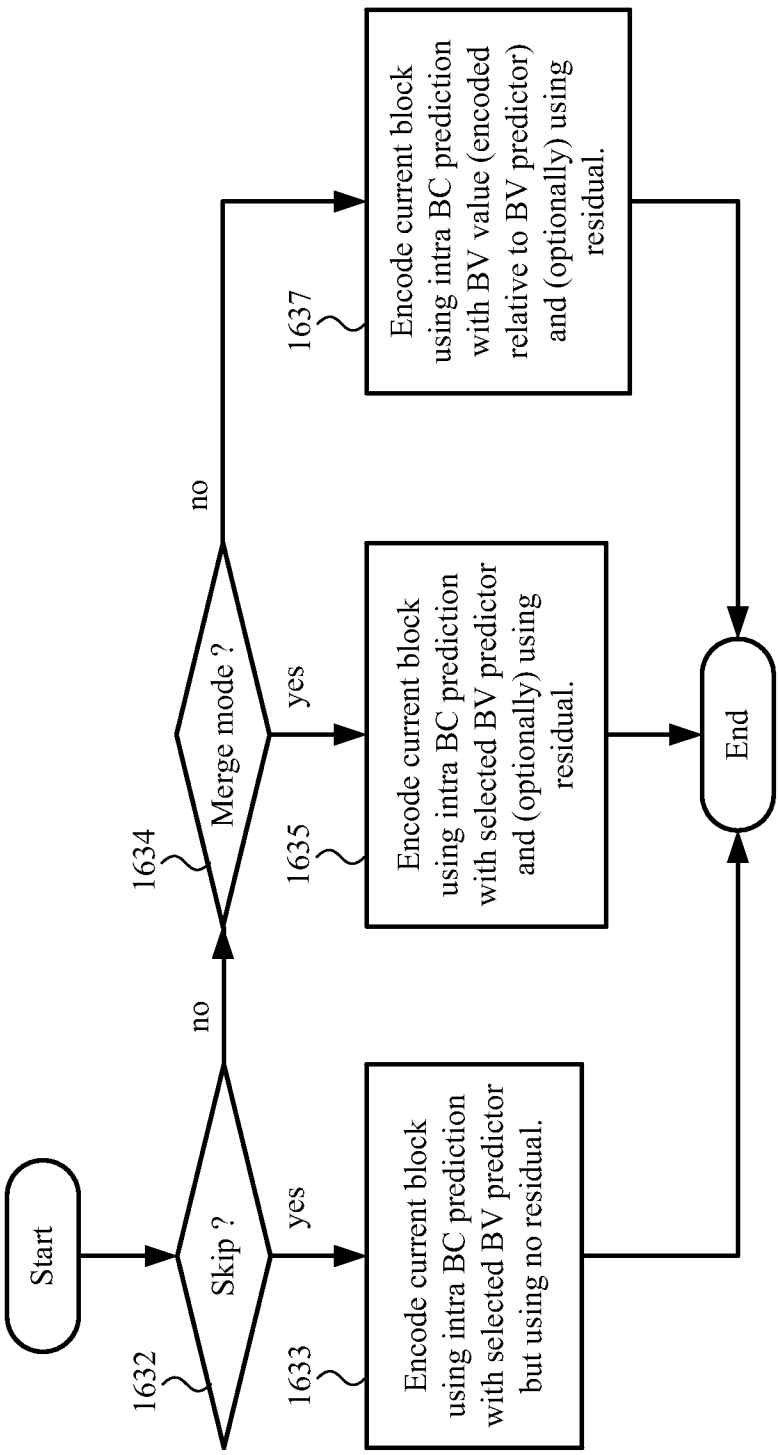
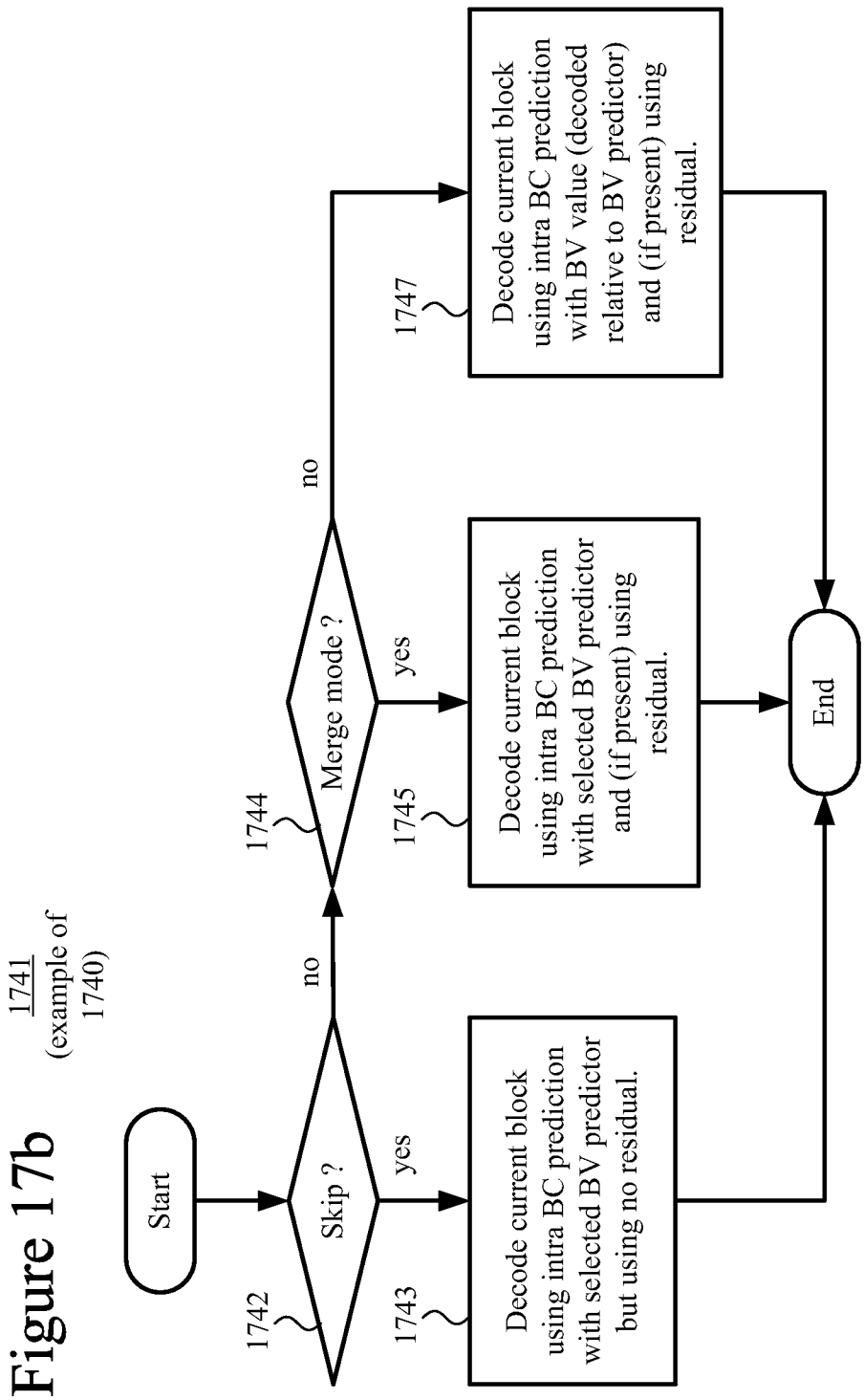

1631
(example of 1630)
Start
1632
Skip ?
yes
no
1633
Encode current block using intra BC prediction with selected BV predictor but using no residual.
1634
Merge mode ?
yes
no
1635
Encode current block using intra BC prediction with selected BV predictor and (optionally) using residual.
1637
Encode current block using intra BC prediction with BV value (encoded relative to BV predictor) and (optionally) using residual.
End

Figure 17b


1741
(example of 1740)
Start
1742
Skip ?
yes
no
1743
Decode current block using intra BC prediction with selected BV predictor but using no residual.
1744
Merge mode ?
yes
no
1745
Decode current block using intra BC prediction with selected BV predictor and (if present) using residual.
1747
Decode current block using intra BC prediction with BV value (decoded relative to BV predictor) and (if present) using residual.
End

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2014/070072

A. CLASSIFICATION OF SUBJECT MATTER

H04N 19/50(2014.01)i

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04N

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

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

DWPI,USTXT,CNTXT: HEVC,block vector,zero,coding,decoding,predict+,encod+,intra block copy,motion vector,candidate, skip mode,indicat+,index,difference

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CN 103392340 A (MEDIATEK INC) 13 November 2013 (2013-11-13) description paragraphs [0005], [0027]-[0030], [0039]-[0041], [0046]-[0047]	1, 3-4, 6-14, 38-39
Y	CN 103392340 A (MEDIATEK INC) 13 November 2013 (2013-11-13) description paragraphs [0005], [0027]-[0030], [0039]-[0041], [0046]-[0047]	2, 5, 20, 34
Y	CN 103430540 A (QUALCOMM INC) 04 December 2013 (2013-12-04) description paragraphs [0049]	2, 5
Y	WO 2007/119198 A1 (KONINK PHILIPS ELECTRONICS NV) 25 October 2007 (2007-10-25) description page 1, line 24 to page 2, line 5	2, 5, 15-20
Y	US 2014/0002599 A1 (SAMSUNG ELECTRONICS CO LTD) 02 January 2014 (2014-01-02) description paragraphs [0059]-[0060], [0113]-[0114]	15-37
Y	WO 2012/174990 A1 (MEDIATEK INC) 27 December 2012 (2012-12-27) description paragraph [0028]	21-37

☑ Further documents are listed in the continuation of Box C. ☑ See patent family annex.

* Special categories of cited documents:

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
25 September 2014	**10 October 2014**

Name and mailing address of the ISA/	Authorized officer
STATE INTELLECTUAL PROPERTY OFFICE OF THE P.R.CHINA(ISA/CN) **6,Xitucheng Rd., Jimen Bridge, Haidian District, Beijing 100088 China**	**SUN,Jiachen**
Facsimile No. **(86-10)62019451**	Telephone No. **(86-10)62411520**

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2014/070072

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2012/0281760 A1 (KIM H J) 08 November 2012 (2012-11-08) the whole document	1-39

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2014/070072

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
CN	103392340	A	13 November 2013	KR	20130108420	A	02 October 2013
				WO	2012122783	A1	20 September 2012
				US	2012236942	A1	20 September 2012
				EP	2687015	A1	22 January 2014
				AU	2011362447	A1	18 July 2013
CN	103430540	A	04 December 2013	WO	2012121808	A1	13 September 2012
				EP	2684356	A1	15 January 2014
				JP	2014511648	A	15 May 2014
				IN	201301660	P3	29 August 2014
				TW	201238358	A	16 September 2012
				US	2012230392	A1	13 September 2012
WO	2007/119198	A1	25 October 2007	US	2009316786	A1	24 December 2009
				EP	2011342	A1	07 January 2009
				CN	101422047	A	29 April 2009
				JP	4997281	B2	08 August 2012
				JP	2009533928	A	17 September 2009
				CN	101422047	B	12 January 2011
US	2014/0002599	A1	02 January 2014	KR	20120080122	A	16 July 2012
				WO	2012093879	A2	12 July 2012
				WO	2012093879	A3	29 November 2012
WO	2012/174990	A1	27 December 2012	US	2014092981	A1	03 April 2014
				KR	20130135368	A	10 December 2013
				EP	2724538	A1	30 April 2014
				CN	103621090	A	05 March 2014
				CA	2830290	A1	27 December 2012
US	2012/0281760	A1	08 November 2012		Non	e	