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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
02 January 2025 (02.01.2025)

WIPO | PCT

(10) International Publication Number
WO 2025/002871 A1

(51) International Patent Classification:
H04N 19/105 (2014.01) *H04N 19/52* (2014.01) *H04N 19/11* (2014.01) *H04N 19/54* (2014.01) *H04N 19/139* (2014.01) *H04N 19/593* (2014.01) *H04N 19/159* (2014.01) *H04N 19/70* (2014.01) *H04N 19/176* (2014.01)

(21) International Application Number: PCT/EP2024/066741

(22) International Filing Date: 17 June 2024 (17.06.2024)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
23306025.0 27 June 2023 (27.06.2023) EP

(71) Applicant: INTERDIGITAL CE PATENT HOLDINGS, SAS [FR/FR]; 3 rue du Colonel Moll, 75017 Paris (FR).

(72) Inventors: LE LEANNEC, Fabrice; c/o InterDigital, Immeuble ZEN 2, 845A Avenue des Champs Blancs, 35510 CESSON-SEVIGNE (FR). GALPIN, Franck; c/o InterDigital, Immeuble ZEN 2, 845A Avenue des Champs Blancs, 35510 CESSON-SEVIGNE (FR). ROBERT, Antoine; c/o InterDigital, Immeuble ZEN 2, 845A Avenue des Champs Blancs, 35510 CESSON-SEVIGNE (FR). NASER, Karam; c/o InterDigital, Immeuble ZEN 2, 845A Avenue des Champs Blancs, 35510 CESSON-SEVIGNE (FR).

(74) Agent: INTERDIGITAL; Immeuble Zen 2, 845A Avenue des Champs Blancs, 35510 CESSON-SEVIGNE (FR).

(54) Title: AFFINE BLOCK VECTOR MODEL FOR INTRA BLOCK COPY

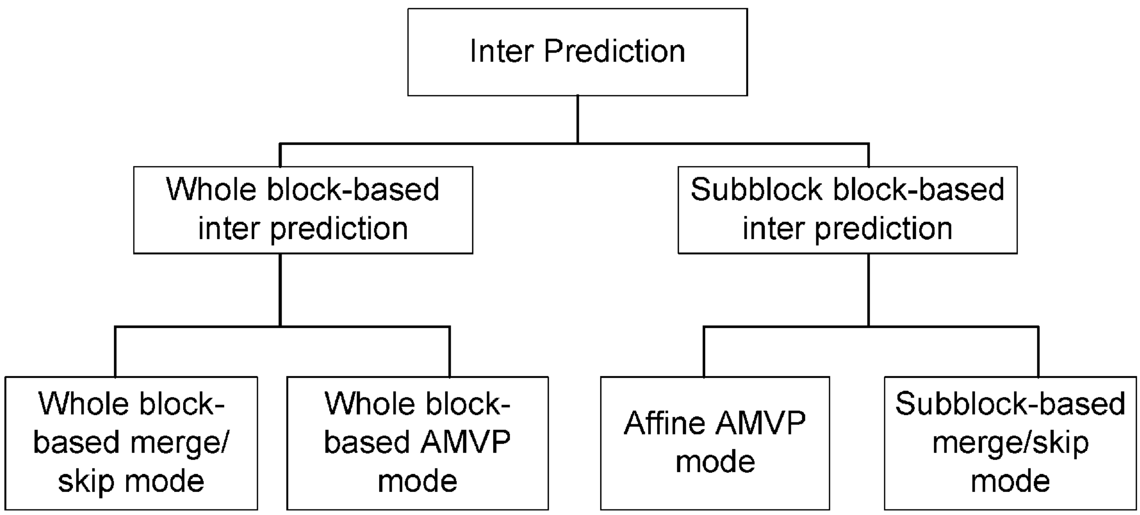

FIG. 8

(57) Abstract: Systems, methods, and instrumentalities are disclosed herein for video encoding and/or video decoding using an intra block copy (IBC) mode. In examples, a device (e.g., a decoder) may obtain an IBC mode indication in video data. The device may determine that a current block is associated with an IBC mode based on the indication. Based on the determination, the device may obtain one or more (e.g., multiple) control point block vectors (CPBVs) associated with the current block. The device may decode the current block based on the CPBVs. In examples, a device (e.g., an encoder) may obtain a current block associated with a video content. The device may determine whether the current block is associated with an affine IBC mode. Based on a determination, the device may obtain one or more (e.g., multiple) CPBVs associated with the current block. The device may encode the current block based on the CPBVs.

WO 2025/002871 A1

[Continued on next page]

(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, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, 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, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, ME, 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))*

AFFINE BLOCK VECTOR MODEL FOR INTRA BLOCK COPY

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of European Provisional Application No. 23306025.9 filed June 27, 2023, the contents of which are incorporated by reference herein.

BACKGROUND

[0002] Video coding systems may be used to compress digital video signals, e.g., to reduce the storage and/or transmission bandwidth needed for such signals. Video coding systems may include, for example, block-based, wavelet-based, and/or object-based systems.

SUMMARY

[0003] Systems, methods, and instrumentalities are disclosed herein for video encoding and/or video decoding using an intra block copy (IBC) mode.

[0004] In examples, a device for video decoding, such as a video decoder, may be configured to obtain an affine IBC mode indication in video data. The device may determine that a current block is associated with an affine IBC, e.g., based on the obtained affine IBC mode indication. Based on the determination that the current block is associated with the affine IBC, the device may obtain one or more (e.g., multiple) control point block vectors (CPBVs) associated with the current block. Based on the one or more (e.g., multiple) CPBVs, the device may decode the current block.

[0005] In examples, a device for video encoding, such as a video encoder, may be configured to obtain a current block associated with a video content. The device may determine whether a current block is associated with an affine IBC. Based on a determination that the current block is associated with the affine IBC, the device may obtain one or more (e.g., multiple) CPBVs associated with the current block. Based on the one or more (e.g., multiple) CPBVs, the device for video encoding may encode the current block.

[0006] In examples, the device may determine that the current block is associated with a merge IBC mode. Based on the determination that the current block is associated with the merge IBC mode, the device may construct an IBC affine control point block vector (BV) prediction candidate list. The one or more (e.g., multiple) CPBVs may be obtained based on the BV prediction candidate list.

[0007] In examples, the device may apply affine motion compensation based on the one or more (e.g., multiple) CPBVs, e.g., to obtain a prediction block. The device may apply prediction refinement based on optical flow (PROF) to the obtained prediction block.

[0008] In examples, the device for video encoding, e.g., the video encoder, may include an indication, such as an affine IBC mode indication, in video data. For example, the video encoder may include an affine IBC mode indication in the video data based on a determination that the current block is associated with the IBC mode.

[0009] In examples, a device, such as a decoder, may be configured to obtain a current block in a current picture. As described herein, the current picture may be or may include camera-captured video content. The camera-captured video content may be referred to interchangeably as natural video and/or natural video content.

[0010] The device may determine that the current block is associated with an IBC mode. For example, the IBC mode may be or may be associated with an affine IBC mode. Based on the determination that the current block is associated with the affine IBC mode, the device may determine that the current block is associated with a merge IBC mode.

[0011] Based on the determination that the current block is associated with the merge IBC mode, the device may construct an IBC affine control block vector prediction candidate list. The device may obtain one or more control point block vectors (CPBVs) associated with the current block. Based on the CPBVs, the device may compute an affine block vector (BV) field associated with the current block. The device may obtain a prediction block associated with the current block. For example, the device may obtain a prediction block based on applying affine motion compensation to the current block. The device may apply prediction refinement based on optical flow (PROF) to the prediction block. The device may add the prediction block and a residual block associated with the current block.

[0012] Based on the determination that the current block is not associated with the merge IBC mode, the device may determine that the current block is associated with an IBC affine advanced motion vector prediction (AMVP) mode. The device may construct an IBC AMVP control block vector prediction candidate list. The device may add a BV difference to corresponding BVs. The BV difference may be associated with the constructed IBC AMVP control block vector prediction candidate list. The device may obtain a CPBV associated with the current block. Based on the CPBV, the device may compute an affine BV field associated with the current block. The device may obtain a prediction block associated with the current block. For example, the device may obtain a prediction block based on applying affine motion compensation to the current block. The device may apply PROF to the prediction block. The device may add the prediction block and a residual block associated with the current block.

[0013] These examples may be performed by a device with a processor. The device may be a decoder. These examples may be performed by a computer program product which is stored on a non-transitory computer readable medium and includes program code instructions. These examples may be performed by a computer program comprising program code instructions.

[0014] Systems, methods, and instrumentalities described herein may involve a decoder. In some examples, the systems, methods, and instrumentalities described herein may involve an encoder. In some examples, the systems, methods, and instrumentalities described herein may involve a signal (e.g., from an encoder and/or received by a decoder). A computer-readable medium may include instructions for causing one or more processors to perform methods described herein. A computer program product may include instructions which, when the program is executed by one or more processors, may cause the one or more processors to carry out the methods described herein.

BRIEF DESCRIPTION OF THE DRAWINGS

[0015] FIG. 1A is a system diagram illustrating an example communications system in which one or more disclosed embodiments may be implemented.

[0016] FIG. 1B is a system diagram illustrating an example wireless transmit/receive unit (WTRU) that may be used within the communications system illustrated in FIG. 1A according to an embodiment.

[0017] FIG. 1C is a system diagram illustrating an example radio access network (RAN) and an example core network (CN) that may be used within the communications system illustrated in FIG. 1A according to an embodiment.

[0018] FIG. 1D is a system diagram illustrating a further example RAN and a further example CN that may be used within the communications system illustrated in FIG. 1A according to an embodiment.

[0019] FIG. 2 illustrates an example video encoder.

[0020] FIG. 3 illustrates an example video decoder.

[0021] FIG. 4 illustrates an example of a system in which various aspects and examples may be implemented.

[0022] FIGs. 5A-5D illustrate examples of intra block copy (IBC) reference regions depending on a current block prediction.

[0023] FIG. 6 illustrates one or more padding candidates for the replacement of a zero-vector in an IBC list.

[0024] FIG. 7 illustrates an example reference area for IBC if a coding tree unit (CTU) (m,n) is coded.

[0025] FIG. 8 illustrates an example of whole-block and sub-block-based motion representation categories.

[0026] FIGs. 9A-9B illustrate an example control point based affine motion models.

[0027] FIG. 10 illustrates an example affine motion field representation on a 4x4 sub-block basis.

[0028] FIG. 11 illustrates example locations of inherited affine motion predictors.

[0029] FIG. 12 illustrates an example control point motion vector (CPMV) inheritance from an affine block to another affine block.

[0030] FIG. 13 illustrates example locations of candidate positions for constructed affine merge mode.

[0031] FIG. 14 illustrates an example sub-block MV V_{SB} and pixel $\Delta v(i,j)$.

[0032] FIG. 15 illustrates an example IBC affine CU decoding procedure.

DETAILED DESCRIPTION

[0033] A more detailed understanding may be had from the following description, given by way of example in conjunction with the accompanying drawings.

[0034] FIG. 1A is a diagram illustrating an example communications system 100 in which one or more disclosed embodiments may be implemented. The communications system 100 may be a multiple access system that provides content, such as voice, data, video, messaging, broadcast, etc., to multiple wireless users. The communications system 100 may enable multiple wireless users to access such content through the sharing of system resources, including wireless bandwidth. For example, the communications systems 100 may employ one or more channel access methods, such as code division multiple access (CDMA), time division multiple access (TDMA), frequency division multiple access (FDMA), orthogonal FDMA (OFDMA), single-carrier FDMA (SC-FDMA), zero-tail unique-word DFT-Spread OFDM (ZT UW DTS-s OFDM), unique word OFDM (UW-OFDM), resource block-filtered OFDM, filter bank multicarrier (FBMC), and the like.

[0035] As shown in FIG. 1A, the communications system 100 may include wireless transmit/receive units (WTRUs) 102a, 102b, 102c, 102d, a RAN 104/113, a CN 106/115, a public switched telephone network (PSTN) 108, the Internet 110, and other networks 112, though it will be appreciated that the disclosed embodiments contemplate any number of WTRUs, base stations, networks, and/or network elements. Each of the WTRUs 102a, 102b, 102c, 102d may be any type of device configured to operate and/or communicate in a wireless environment. By way of example, the WTRUs 102a, 102b, 102c, 102d, any of which may be referred to as a “station” and/or a “STA”, may be configured to transmit and/or receive wireless signals and may include a user equipment (UE), a mobile station, a fixed or mobile subscriber unit, a subscription-based unit, a pager, a cellular telephone, a personal digital assistant (PDA), a smartphone, a laptop, a netbook, a personal computer, a wireless sensor, a hotspot or Mi-Fi device, an Internet of Things (IoT) device, a watch or other wearable, a head-mounted display (HMD), a vehicle, a drone, a medical device and applications (e.g., remote surgery), an industrial device and applications (e.g., a robot and/or other wireless devices operating in an industrial and/or an automated processing chain contexts), a consumer electronics device, a device operating on commercial and/or industrial wireless networks, and the like. Any of the WTRUs 102a, 102b, 102c and 102d may be interchangeably referred to as a UE.

[0036] The communications systems 100 may also include a base station 114a and/or a base station 114b. Each of the base stations 114a, 114b may be any type of device configured to wirelessly interface with at least one of the WTRUs 102a, 102b, 102c, 102d to facilitate access to one or more communication networks, such as the CN 106/115, the Internet 110, and/or the other networks 112. By way of example, the base stations 114a, 114b may be a base transceiver station (BTS), a Node-B, an eNode B, a Home Node B, a Home eNode B, a gNB, a NR NodeB, a site controller, an access point (AP), a wireless router, and the like. While the base stations 114a, 114b are each depicted as a single element, it will be appreciated that the base stations 114a, 114b may include any number of interconnected base stations and/or network elements.

[0037] The base station 114a may be part of the RAN 104/113, which may also include other base stations and/or network elements (not shown), such as a base station controller (BSC), a radio network controller (RNC), relay nodes, etc. The base station 114a and/or the base station 114b may be configured to transmit and/or receive wireless signals on one or more carrier frequencies, which may be referred to as a cell (not shown). These frequencies may be in licensed spectrum, unlicensed spectrum, or a combination of licensed and unlicensed spectrum. A cell may provide coverage for a wireless service to a specific geographical area that may be relatively fixed or that may change over time. The cell may further be divided into cell sectors. For example, the cell associated with the base station 114a may be divided into three sectors. Thus, in one embodiment, the base station 114a may include three transceivers, i.e., one for each sector of the cell. In an embodiment, the base station 114a may employ multiple-input multiple output (MIMO) technology and may utilize multiple transceivers for each sector of the cell. For example, beamforming may be used to transmit and/or receive signals in desired spatial directions.

[0038] The base stations 114a, 114b may communicate with one or more of the WTRUs 102a, 102b, 102c, 102d over an air interface 116, which may be any suitable wireless communication link (e.g., radio frequency (RF), microwave, centimeter wave, micrometer wave, infrared (IR), ultraviolet (UV), visible light, etc.). The air interface 116 may be established using any suitable radio access technology (RAT).

[0039] More specifically, as noted above, the communications system 100 may be a multiple access system and may employ one or more channel access schemes, such as CDMA, TDMA, FDMA, OFDMA, SC-FDMA, and the like. For example, the base station 114a in the RAN 104/113 and the WTRUs 102a, 102b, 102c may implement a radio technology such as Universal Mobile Telecommunications System (UMTS) Terrestrial Radio Access (UTRA), which may establish the air interface 116 using wideband CDMA (WCDMA). WCDMA may include communication protocols such as High-Speed Packet Access (HSPA) and/or Evolved HSPA (HSPA+). HSPA may include High-Speed Downlink (DL) Packet Access (HSDPA) and/or High-Speed UL Packet Access (HSUPA).

[0040] In an embodiment, the base station 114a and the WTRUs 102a, 102b, 102c may implement a radio technology such as Evolved UMTS Terrestrial Radio Access (E-UTRA), which may establish the air interface 116 using Long Term Evolution (LTE) and/or LTE-Advanced (LTE-A) and/or LTE-Advanced Pro (LTE-A Pro).

[0041] In an embodiment, the base station 114a and the WTRUs 102a, 102b, 102c may implement a radio technology such as NR Radio Access, which may establish the air interface 116 using New Radio (NR).

[0042] In an embodiment, the base station 114a and the WTRUs 102a, 102b, 102c may implement multiple radio access technologies. For example, the base station 114a and the WTRUs 102a, 102b, 102c may implement LTE radio access and NR radio access together, for instance using dual connectivity (DC) principles. Thus, the air interface utilized by WTRUs 102a, 102b, 102c may be characterized by multiple types of radio access technologies and/or transmissions sent to/from multiple types of base stations (e.g., an eNB and a gNB).

[0043] In other embodiments, the base station 114a and the WTRUs 102a, 102b, 102c may implement radio technologies such as IEEE 802.11 (i.e., Wireless Fidelity (WiFi), IEEE 802.16 (i.e., Worldwide Interoperability for Microwave Access (WiMAX)), CDMA2000, CDMA2000 1X, CDMA2000 EV-DO, Interim Standard 2000 (IS-2000), Interim Standard 95 (IS-95), Interim Standard 856 (IS-856), Global System for Mobile communications (GSM), Enhanced Data rates for GSM Evolution (EDGE), GSM EDGE (GERAN), and the like.

[0044] The base station 114b in FIG. 1A may be a wireless router, Home Node B, Home eNode B, or access point, for example, and may utilize any suitable RAT for facilitating wireless connectivity in a localized area, such as a place of business, a home, a vehicle, a campus, an industrial facility, an air corridor (e.g., for use by drones), a roadway, and the like. In one embodiment, the base station 114b and the WTRUs 102c, 102d may implement a radio technology such as IEEE 802.11 to establish a wireless local area network (WLAN). In an embodiment, the base station 114b and the WTRUs 102c, 102d may implement a radio technology such as IEEE 802.15 to establish a wireless personal area network (WPAN). In yet another embodiment, the base station 114b and the WTRUs 102c, 102d may utilize a cellular-based RAT (e.g., WCDMA, CDMA2000, GSM, LTE, LTE-A, LTE-A Pro, NR etc.) to establish a picocell or femtocell. As shown in FIG. 1A, the base station 114b may have a direct connection to the Internet 110. Thus, the base station 114b may not be required to access the Internet 110 via the CN 106/115.

[0045] The RAN 104/113 may be in communication with the CN 106/115, which may be any type of network configured to provide voice, data, applications, and/or voice over internet protocol (VoIP) services to one or more of the WTRUs 102a, 102b, 102c, 102d. The data may have varying quality of service (QoS) requirements, such as differing throughput requirements, latency requirements, error tolerance requirements, reliability requirements, data throughput requirements, mobility requirements, and the like. The CN 106/115 may provide call control, billing services, mobile location-based services, pre-paid calling, Internet connectivity, video distribution, etc., and/or perform high-level security functions, such as user authentication. Although not shown in FIG. 1A, it will be appreciated that the RAN 104/113 and/or the CN 106/115 may be in direct or indirect communication with other RANs that employ the same RAT as the RAN 104/113 or a different RAT. For example, in addition to being connected to the RAN 104/113, which may be utilizing a NR radio technology, the CN 106/115 may also be in communication with another RAN (not shown) employing a GSM, UMTS, CDMA 2000, WiMAX, E-UTRA, or WiFi radio technology.

[0046] The CN 106/115 may also serve as a gateway for the WTRUs 102a, 102b, 102c, 102d to access the PSTN 108, the Internet 110, and/or the other networks 112. The PSTN 108 may include circuit-switched telephone networks that provide plain old telephone service (POTS). The Internet 110 may include a global system of interconnected computer networks and devices that use common communication protocols, such as the transmission control protocol (TCP), user datagram protocol (UDP) and/or the internet protocol (IP) in the TCP/IP internet protocol suite. The networks 112 may include wired and/or wireless communications networks owned and/or operated by other service providers. For example, the networks 112 may include another CN connected to one or more RANs, which may employ the same RAT as the RAN 104/113 or a different RAT.

[0047] Some or all of the WTRUs 102a, 102b, 102c, 102d in the communications system 100 may include multi-mode capabilities (e.g., the WTRUs 102a, 102b, 102c, 102d may include multiple transceivers for communicating with different wireless networks over different wireless links). For example, the WTRU 102c shown in FIG. 1A may be configured to communicate with the base station 114a, which may employ a cellular-based radio technology, and with the base station 114b, which may employ an IEEE 802 radio technology.

[0048] FIG. 1B is a system diagram illustrating an example WTRU 102. As shown in FIG. 1B, the WTRU 102 may include a processor 118, a transceiver 120, a transmit/receive element 122, a speaker/microphone 124, a keypad 126, a display/touchpad 128, non-removable memory 130, removable memory 132, a power source 134, a global positioning system (GPS) chipset 136, and/or other peripherals

138, among others. It will be appreciated that the WTRU 102 may include any sub-combination of the foregoing elements while remaining consistent with an embodiment.

[0049] The processor 118 may be a general purpose processor, a special purpose processor, a conventional processor, a digital signal processor (DSP), a plurality of microprocessors, one or more microprocessors in association with a DSP core, a controller, a microcontroller, Application Specific Integrated Circuits (ASICs), Field Programmable Gate Arrays (FPGAs) circuits, any other type of integrated circuit (IC), a state machine, and the like. As suggested above, the processor 118 may include a plurality of processors. The processor 118 may perform signal coding, data processing, power control, input/output processing, and/or any other functionality that enables the WTRU 102 to operate in a wireless environment. The processor 118 may be coupled to the transceiver 120, which may be coupled to the transmit/receive element 122. While FIG. 1B depicts the processor 118 and the transceiver 120 as separate components, it will be appreciated that the processor 118 and the transceiver 120 may be integrated together in an electronic package or chip.

[0050] The transmit/receive element 122 may be configured to transmit signals to, or receive signals from, a base station (e.g., the base station 114a) over the air interface 116. For example, in one embodiment, the transmit/receive element 122 may be an antenna configured to transmit and/or receive RF signals. In an embodiment, the transmit/receive element 122 may be an emitter/detector configured to transmit and/or receive IR, UV, or visible light signals, for example. In yet another embodiment, the transmit/receive element 122 may be configured to transmit and/or receive both RF and light signals. It will be appreciated that the transmit/receive element 122 may be configured to transmit and/or receive any combination of wireless signals.

[0051] Although the transmit/receive element 122 is depicted in FIG. 1B as a single element, the WTRU 102 may include any number of transmit/receive elements 122. More specifically, the WTRU 102 may employ MIMO technology. Thus, in one embodiment, the WTRU 102 may include two or more transmit/receive elements 122 (e.g., multiple antennas) for transmitting and receiving wireless signals over the air interface 116.

[0052] The transceiver 120 may be configured to modulate the signals that are to be transmitted by the transmit/receive element 122 and to demodulate the signals that are received by the transmit/receive element 122. As noted above, the WTRU 102 may have multi-mode capabilities. Thus, the transceiver 120 may include multiple transceivers for enabling the WTRU 102 to communicate via multiple RATs, such as NR and IEEE 802.11, for example.

[0053] The processor 118 of the WTRU 102 may be coupled to, and may receive user input data from, the speaker/microphone 124, the keypad 126, and/or the display/touchpad 128 (e.g., a liquid crystal display (LCD) display unit or organic light-emitting diode (OLED) display unit). The processor 118 may also output user data to the speaker/microphone 124, the keypad 126, and/or the display/touchpad 128. In addition, the processor 118 may access information from, and store data in, any type of suitable memory, such as the non-removable memory 130 and/or the removable memory 132. The non-removable memory 130 may include random-access memory (RAM), read-only memory (ROM), a hard disk, or any other type of memory storage device. The removable memory 132 may include a subscriber identity module (SIM) card, a memory stick, a secure digital (SD) memory card, and the like. In other embodiments, the processor 118 may access information from, and store data in, memory that is not physically located on the WTRU 102, such as on a server or a home computer (not shown).

[0054] The processor 118 may receive power from the power source 134, and may be configured to distribute and/or control the power to the other components in the WTRU 102. The power source 134 may be any suitable device for powering the WTRU 102. For example, the power source 134 may include one or more dry cell batteries (e.g., nickel-cadmium (NiCd), nickel-zinc (NiZn), nickel metal hydride (NiMH), lithium-ion (Li-ion), etc.), solar cells, fuel cells, and the like.

[0055] The processor 118 may also be coupled to the GPS chipset 136, which may be configured to provide location information (e.g., longitude and latitude) regarding the current location of the WTRU 102. In addition to, or in lieu of, the information from the GPS chipset 136, the WTRU 102 may receive location information over the air interface 116 from a base station (e.g., base stations 114a, 114b) and/or determine its location based on the timing of the signals being received from two or more nearby base stations. It will be appreciated that the WTRU 102 may acquire location information by way of any suitable location-determination method while remaining consistent with an embodiment.

[0056] The processor 118 may further be coupled to other peripherals 138, which may include one or more software and/or hardware modules that provide additional features, functionality and/or wired or wireless connectivity. For example, the peripherals 138 may include an accelerometer, an e-compass, a satellite transceiver, a digital camera (for photographs and/or video), a universal serial bus (USB) port, a vibration device, a television transceiver, a hands free headset, a Bluetooth® module, a frequency modulated (FM) radio unit, a digital music player, a media player, a video game player module, an Internet browser, a Virtual Reality and/or Augmented Reality (VR/AR) device, an activity tracker, and the like. The peripherals 138 may include one or more sensors, the sensors may be one or more of a gyroscope, an accelerometer, a hall effect sensor, a magnetometer, an orientation sensor, a proximity sensor, a temperature sensor, a time sensor; a geolocation sensor; an altimeter, a light sensor, a touch sensor, a magnetometer, a barometer, a gesture sensor, a biometric sensor, and/or a humidity sensor.

[0057] The WTRU 102 may include a full duplex radio for which transmission and reception of some or all of the signals (e.g., associated with particular subframes for both the UL (e.g., for transmission) and downlink (e.g., for reception) may be concurrent and/or simultaneous. The full duplex radio may include an interference management unit to reduce and or substantially eliminate self-interference via either hardware (e.g., a choke) or signal processing via a processor (e.g., a separate processor (not shown) or via processor 118). In an embodiment, the WRTU 102 may include a half-duplex radio for which transmission and reception of some or all of the signals (e.g., associated with particular subframes for either the UL (e.g., for transmission) or the downlink (e.g., for reception)).

[0058] FIG. 1C is a system diagram illustrating the RAN 104 and the CN 106 according to an embodiment. As noted above, the RAN 104 may employ an E-UTRA radio technology to communicate with the WTRUs 102a, 102b, 102c over the air interface 116. The RAN 104 may also be in communication with the CN 106.

[0059] The RAN 104 may include eNode-Bs 160a, 160b, 160c, though it will be appreciated that the RAN 104 may include any number of eNode-Bs while remaining consistent with an embodiment. The eNode-Bs 160a, 160b, 160c may each include one or more transceivers for communicating with the WTRUs 102a, 102b, 102c over the air interface 116. In one embodiment, the eNode-Bs 160a, 160b, 160c may implement MIMO technology. Thus, the eNode-B 160a, for example, may use multiple antennas to transmit wireless signals to, and/or receive wireless signals from, the WTRU 102a.

[0060] Each of the eNode-Bs 160a, 160b, 160c may be associated with a particular cell (not shown) and may be configured to handle radio resource management decisions, handover decisions, scheduling of users in the UL and/or DL, and the like. As shown in FIG. 1C, the eNode-Bs 160a, 160b, 160c may communicate with one another over an X2 interface.

[0061] The CN 106 shown in FIG. 1C may include a mobility management entity (MME) 162, a serving gateway (SGW) 164, and a packet data network (PDN) gateway (or PGW) 166. While each of the foregoing elements are depicted as part of the CN 106, it will be appreciated that any of these elements may be owned and/or operated by an entity other than the CN operator.

[0062] The MME 162 may be connected to each of the eNode-Bs 160a, 160b, 160c in the RAN 104 via an S1 interface and may serve as a control node. For example, the MME 162 may be responsible for authenticating users of the WTRUs 102a, 102b, 102c, bearer activation/deactivation, selecting a particular serving gateway during an initial attach of the WTRUs 102a, 102b, 102c, and the like. The MME 162 may provide a control plane function for switching between the RAN 104 and other RANs (not shown) that employ other radio technologies, such as GSM and/or WCDMA.

[0063] The SGW 164 may be connected to each of the eNode Bs 160a, 160b, 160c in the RAN 104 via the S1 interface. The SGW 164 may generally route and forward user data packets to/from the WTRUs 102a, 102b, 102c. The SGW 164 may perform other functions, such as anchoring user planes during inter-eNode B handovers, triggering paging when DL data is available for the WTRUs 102a, 102b, 102c, managing and storing contexts of the WTRUs 102a, 102b, 102c, and the like.

[0064] The SGW 164 may be connected to the PGW 166, which may provide the WTRUs 102a, 102b, 102c with access to packet-switched networks, such as the Internet 110, to facilitate communications between the WTRUs 102a, 102b, 102c and IP-enabled devices.

[0065] The CN 106 may facilitate communications with other networks. For example, the CN 106 may provide the WTRUs 102a, 102b, 102c with access to circuit-switched networks, such as the PSTN 108, to facilitate communications between the WTRUs 102a, 102b, 102c and traditional land-line communications devices. For example, the CN 106 may include, or may communicate with, an IP gateway (e.g., an IP multimedia subsystem (IMS) server) that serves as an interface between the CN 106 and the PSTN 108. In addition, the CN 106 may provide the WTRUs 102a, 102b, 102c with access to the other networks 112, which may include other wired and/or wireless networks that are owned and/or operated by other service providers.

[0066] Although the WTRU is described in FIGS. 1A-1D as a wireless terminal, it is contemplated that in certain representative embodiments that such a terminal may use (e.g., temporarily or permanently) wired communication interfaces with the communication network.

[0067] In representative embodiments, the other network 112 may be a WLAN.

[0068] A WLAN in Infrastructure Basic Service Set (BSS) mode may have an Access Point (AP) for the BSS and one or more stations (STAs) associated with the AP. The AP may have an access or an interface to a Distribution System (DS) or another type of wired/wireless network that carries traffic in to and/or out of the BSS. Traffic to STAs that originates from outside the BSS may arrive through the AP and may be delivered to the STAs. Traffic originating from STAs to destinations outside the BSS may be sent to the AP to be delivered to respective destinations. Traffic between STAs within the BSS may be sent through the AP, for example, where the source STA may send traffic to the AP and the AP may deliver the traffic to the destination STA. The traffic between STAs within a BSS may be considered and/or referred to as peer-to-peer traffic. The peer-to-peer traffic may be sent between (e.g., directly between) the source and destination STAs with a direct link setup (DLS). In certain representative embodiments, the DLS may use an 802.11e DLS or an 802.11z tunneled DLS (TDLS). A WLAN using an Independent BSS (IBSS) mode may not have an AP, and the STAs (e.g., all of the STAs) within or using the IBSS may communicate directly with each other. The IBSS mode of communication may sometimes be referred to herein as an “ad-hoc” mode of communication.

[0069] When using the 802.11ac infrastructure mode of operation or a similar mode of operations, the AP may transmit a beacon on a fixed channel, such as a primary channel. The primary channel may be a fixed width (e.g., 20 MHz wide bandwidth) or a dynamically set width via signaling. The primary channel may be the operating channel of the BSS and may be used by the STAs to establish a connection with the AP. In certain representative embodiments, Carrier Sense Multiple Access with Collision Avoidance

(CSMA/CA) may be implemented, for example in in 802.11 systems. For CSMA/CA, the STAs (e.g., every STA), including the AP, may sense the primary channel. If the primary channel is sensed/detected and/or determined to be busy by a particular STA, the particular STA may back off. One STA (e.g., only one station) may transmit at any given time in a given BSS.

[0070] High Throughput (HT) STAs may use a 40 MHz wide channel for communication, for example, via a combination of the primary 20 MHz channel with an adjacent or nonadjacent 20 MHz channel to form a 40 MHz wide channel.

[0071] Very High Throughput (VHT) STAs may support 20MHz, 40 MHz, 80 MHz, and/or 160 MHz wide channels. The 40 MHz, and/or 80 MHz, channels may be formed by combining contiguous 20 MHz channels. A 160 MHz channel may be formed by combining 8 contiguous 20 MHz channels, or by combining two non-contiguous 80 MHz channels, which may be referred to as an 80+80 configuration. For the 80+80 configuration, the data, after channel encoding, may be passed through a segment parser that may divide the data into two streams. Inverse Fast Fourier Transform (IFFT) processing, and time domain processing, may be done on each stream separately. The streams may be mapped on to the two 80 MHz channels, and the data may be transmitted by a transmitting STA. At the receiver of the receiving STA, the above described operation for the 80+80 configuration may be reversed, and the combined data may be sent to the Medium Access Control (MAC).

[0072] Sub 1 GHz modes of operation are supported by 802.11af and 802.11ah. The channel operating bandwidths, and carriers, are reduced in 802.11af and 802.11ah relative to those used in 802.11n, and 802.11ac. 802.11af supports 5 MHz, 10 MHz and 20 MHz bandwidths in the TV White Space (TVWS) spectrum, and 802.11ah supports 1 MHz, 2 MHz, 4 MHz, 8 MHz, and 16 MHz bandwidths using non-TVWS spectrum. According to a representative embodiment, 802.11ah may support Meter Type Control/Machine-Type Communications, such as MTC devices in a macro coverage area. MTC devices may have certain capabilities, for example, limited capabilities including support for (e.g., only support for) certain and/or limited bandwidths. The MTC devices may include a battery with a battery life above a threshold (e.g., to maintain a very long battery life).

[0073] WLAN systems, which may support multiple channels, and channel bandwidths, such as 802.11n, 802.11ac, 802.11af, and 802.11ah, include a channel which may be designated as the primary channel. The primary channel may have a bandwidth equal to the largest common operating bandwidth supported by all STAs in the BSS. The bandwidth of the primary channel may be set and/or limited by a STA, from among all STAs in operating in a BSS, which supports the smallest bandwidth operating mode. In the example of 802.11ah, the primary channel may be 1 MHz wide for STAs (e.g., MTC type devices) that support (e.g., only support) a 1 MHz mode, even if the AP, and other STAs in the BSS support 2 MHz, 4 MHz, 8 MHz, 16 MHz, and/or other channel bandwidth operating modes. Carrier sensing and/or Network

Allocation Vector (NAV) settings may depend on the status of the primary channel. If the primary channel is busy, for example, due to a STA (which supports only a 1 MHz operating mode), transmitting to the AP, the entire available frequency bands may be considered busy even though a majority of the frequency bands remains idle and may be available.

[0074] In the United States, the available frequency bands, which may be used by 802.11ah, are from 902 MHz to 928 MHz. In Korea, the available frequency bands are from 917.5 MHz to 923.5 MHz. In Japan, the available frequency bands are from 916.5 MHz to 927.5 MHz. The total bandwidth available for 802.11ah is 6 MHz to 26 MHz depending on the country code.

[0075] FIG. 1D is a system diagram illustrating the RAN 113 and the CN 115 according to an embodiment. As noted above, the RAN 113 may employ an NR radio technology to communicate with the WTRUs 102a, 102b, 102c over the air interface 116. The RAN 113 may also be in communication with the CN 115.

[0076] The RAN 113 may include gNBs 180a, 180b, 180c, though it will be appreciated that the RAN 113 may include any number of gNBs while remaining consistent with an embodiment. The gNBs 180a, 180b, 180c may each include one or more transceivers for communicating with the WTRUs 102a, 102b, 102c over the air interface 116. In one embodiment, the gNBs 180a, 180b, 180c may implement MIMO technology. For example, gNBs 180a, 108b may utilize beamforming to transmit signals to and/or receive signals from the gNBs 180a, 180b, 180c. Thus, the gNB 180a, for example, may use multiple antennas to transmit wireless signals to, and/or receive wireless signals from, the WTRU 102a. In an embodiment, the gNBs 180a, 180b, 180c may implement carrier aggregation technology. For example, the gNB 180a may transmit multiple component carriers to the WTRU 102a (not shown). A subset of these component carriers may be on unlicensed spectrum while the remaining component carriers may be on licensed spectrum. In an embodiment, the gNBs 180a, 180b, 180c may implement Coordinated Multi-Point (CoMP) technology. For example, WTRU 102a may receive coordinated transmissions from gNB 180a and gNB 180b (and/or gNB 180c).

[0077] The WTRUs 102a, 102b, 102c may communicate with gNBs 180a, 180b, 180c using transmissions associated with a scalable numerology. For example, the OFDM symbol spacing and/or OFDM subcarrier spacing may vary for different transmissions, different cells, and/or different portions of the wireless transmission spectrum. The WTRUs 102a, 102b, 102c may communicate with gNBs 180a, 180b, 180c using subframe or transmission time intervals (TTIs) of various or scalable lengths (e.g., containing varying number of OFDM symbols and/or lasting varying lengths of absolute time).

[0078] The gNBs 180a, 180b, 180c may be configured to communicate with the WTRUs 102a, 102b, 102c in a standalone configuration and/or a non-standalone configuration. In the standalone configuration, WTRUs 102a, 102b, 102c may communicate with gNBs 180a, 180b, 180c without also accessing other

RANs (e.g., such as eNode-Bs 160a, 160b, 160c). In the standalone configuration, WTRUs 102a, 102b, 102c may utilize one or more of gNBs 180a, 180b, 180c as a mobility anchor point. In the standalone configuration, WTRUs 102a, 102b, 102c may communicate with gNBs 180a, 180b, 180c using signals in an unlicensed band. In a non-standalone configuration WTRUs 102a, 102b, 102c may communicate with/connect to gNBs 180a, 180b, 180c while also communicating with/connecting to another RAN such as eNode-Bs 160a, 160b, 160c. For example, WTRUs 102a, 102b, 102c may implement DC principles to communicate with one or more gNBs 180a, 180b, 180c and one or more eNode-Bs 160a, 160b, 160c substantially simultaneously. In the non-standalone configuration, eNode-Bs 160a, 160b, 160c may serve as a mobility anchor for WTRUs 102a, 102b, 102c and gNBs 180a, 180b, 180c may provide additional coverage and/or throughput for servicing WTRUs 102a, 102b, 102c.

[0079] Each of the gNBs 180a, 180b, 180c may be associated with a particular cell (not shown) and may be configured to handle radio resource management decisions, handover decisions, scheduling of users in the UL and/or DL, support of network slicing, dual connectivity, interworking between NR and E-UTRA, routing of user plane data towards User Plane Function (UPF) 184a, 184b, routing of control plane information towards Access and Mobility Management Function (AMF) 182a, 182b and the like. As shown in FIG. 1D, the gNBs 180a, 180b, 180c may communicate with one another over an Xn interface.

[0080] The CN 115 shown in FIG. 1D may include at least one AMF 182a, 182b, at least one UPF 184a,184b, at least one Session Management Function (SMF) 183a, 183b, and possibly a Data Network (DN) 185a, 185b. While each of the foregoing elements are depicted as part of the CN 115, it will be appreciated that any of these elements may be owned and/or operated by an entity other than the CN operator.

[0081] The AMF 182a, 182b may be connected to one or more of the gNBs 180a, 180b, 180c in the RAN 113 via an N2 interface and may serve as a control node. For example, the AMF 182a, 182b may be responsible for authenticating users of the WTRUs 102a, 102b, 102c, support for network slicing (e.g., handling of different PDU sessions with different requirements), selecting a particular SMF 183a, 183b, management of the registration area, termination of NAS signaling, mobility management, and the like. Network slicing may be used by the AMF 182a, 182b in order to customize CN support for WTRUs 102a, 102b, 102c based on the types of services being utilized WTRUs 102a, 102b, 102c. For example, different network slices may be established for different use cases such as services relying on ultra-reliable low latency (URLLC) access, services relying on enhanced massive mobile broadband (eMBB) access, services for machine type communication (MTC) access, and/or the like. The AMF 162 may provide a control plane function for switching between the RAN 113 and other RANs (not shown) that employ other radio technologies, such as LTE, LTE-A, LTE-A Pro, and/or non-3GPP access technologies such as WiFi.

[0082] The SMF 183a, 183b may be connected to an AMF 182a, 182b in the CN 115 via an N11 interface. The SMF 183a, 183b may also be connected to a UPF 184a, 184b in the CN 115 via an N4 interface. The SMF 183a, 183b may select and control the UPF 184a, 184b and configure the routing of traffic through the UPF 184a, 184b. The SMF 183a, 183b may perform other functions, such as managing and allocating UE IP address, managing PDU sessions, controlling policy enforcement and QoS, providing downlink data notifications, and the like. A PDU session type may be IP-based, non-IP based, Ethernet-based, and the like.

[0083] The UPF 184a, 184b may be connected to one or more of the gNBs 180a, 180b, 180c in the RAN 113 via an N3 interface, which may provide the WTRUs 102a, 102b, 102c with access to packet-switched networks, such as the Internet 110, to facilitate communications between the WTRUs 102a, 102b, 102c and IP-enabled devices. The UPF 184a, 184b may perform other functions, such as routing and forwarding packets, enforcing user plane policies, supporting multi-homed PDU sessions, handling user plane QoS, buffering downlink packets, providing mobility anchoring, and the like.

[0084] The CN 115 may facilitate communications with other networks. For example, the CN 115 may include, or may communicate with, an IP gateway (e.g., an IP multimedia subsystem (IMS) server) that serves as an interface between the CN 115 and the PSTN 108. In addition, the CN 115 may provide the WTRUs 102a, 102b, 102c with access to the other networks 112, which may include other wired and/or wireless networks that are owned and/or operated by other service providers. In one embodiment, the WTRUs 102a, 102b, 102c may be connected to a local Data Network (DN) 185a, 185b through the UPF 184a, 184b via the N3 interface to the UPF 184a, 184b and an N6 interface between the UPF 184a, 184b and the DN 185a, 185b.

[0085] In view of Figures 1A-1D, and the corresponding description of Figures 1A-1D, one or more, or all, of the functions described herein with regard to one or more of: WTRU 102a-d, Base Station 114a-b, eNode-B 160a-c, MME 162, SGW 164, PGW 166, gNB 180a-c, AMF 182a-b, UPF 184a-b, SMF 183a-b, DN 185a-b, and/or any other device(s) described herein, may be performed by one or more emulation devices (not shown). The emulation devices may be one or more devices configured to emulate one or more, or all, of the functions described herein. For example, the emulation devices may be used to test other devices and/or to simulate network and/or WTRU functions.

[0086] The emulation devices may be designed to implement one or more tests of other devices in a lab environment and/or in an operator network environment. For example, the one or more emulation devices may perform the one or more, or all, functions while being fully or partially implemented and/or deployed as part of a wired and/or wireless communication network in order to test other devices within the communication network. The one or more emulation devices may perform the one or more, or all, functions while being temporarily implemented/deployed as part of a wired and/or wireless communication network.

The emulation device may be directly coupled to another device for purposes of testing and/or may perform testing using over-the-air wireless communications.

[0087] The one or more emulation devices may perform the one or more, including all, functions while not being implemented/deployed as part of a wired and/or wireless communication network. For example, the emulation devices may be utilized in a testing scenario in a testing laboratory and/or a non-deployed (e.g., testing) wired and/or wireless communication network in order to implement testing of one or more components. The one or more emulation devices may be testing equipment. Direct RF coupling and/or wireless communications via RF circuitry (e.g., which may include one or more antennas) may be used by the emulation devices to transmit and/or receive data.

[0088] This application describes a variety of aspects, including tools, features, examples, models, approaches, etc. Many of these aspects are described with specificity and, at least to show the individual characteristics, are often described in a manner that may sound limiting. However, this is for purposes of clarity in description, and does not limit the application or scope of those aspects. Indeed, all of the different aspects may be combined and interchanged to provide further aspects. Moreover, the aspects may be combined and interchanged with aspects described in earlier filings as well.

[0089] The aspects described and contemplated in this application may be implemented in many different forms. FIGs. 5-15 described herein may provide some examples, but other examples are contemplated. The discussion of FIGs. 5-15 does not limit the breadth of the implementations. At least one of the aspects generally relates to video encoding and decoding, and at least one other aspect generally relates to transmitting a bitstream generated or encoded. These and other aspects may be implemented as a method, an apparatus, a computer readable storage medium having stored thereon instructions for encoding or decoding video data according to any of the methods described, and/or a computer readable storage medium having stored thereon a bitstream generated according to any of the methods described.

[0090] In the present application, the terms “reconstructed” and “decoded” may be used interchangeably, the terms “pixel” and “sample” may be used interchangeably, and the terms “image,” “picture” and “frame” may be used interchangeably.

[0091] Various methods are described herein, and each of the methods comprises one or more steps or actions for achieving the described method. Unless a specific order of steps or actions is required for the proper operation of the method, the order and/or use of specific steps and/or actions may be modified or combined. Additionally, terms such as “first,” “second,” etc. may be used in various examples to modify an element, component, step, operation, etc., such as, for example, a “first decoding” and a “second decoding.” Use of such terms does not imply an ordering to the modified operations unless specifically required. So, in this example, the first decoding need not be performed before the second decoding, and may occur, for example, before, during, or in an overlapping time period with the second decoding.

[0092] Various methods and other aspects described in this application may be used to modify modules, for example, decoding modules, of a video encoder (200) and a video decoder (300) as shown in FIG. 2 and FIG. 3. Moreover, the subject matter disclosed herein may be applied, for example, to any type, format or version of video coding, whether described in a standard or a recommendation, whether pre-existing or future-developed, and extensions of any such standards and recommendations. Unless indicated otherwise, or technically precluded, the aspects described in this application may be used individually or in combination.

[0093] Various numeric values are used in examples described in the present application, such as bits, bit depth, etc. These and other specific values are for purposes of describing examples and the aspects described are not limited to these specific values.

[0094] FIG. 2 illustrates a diagram showing an example video encoder. Variations of example encoder 200 are contemplated, but the encoder 200 is described below for purposes of clarity without describing all expected variations.

[0095] Before being encoded, the video sequence may go through pre-encoding processing (201), for example, applying a color transform to the input color picture (e.g., conversion from RGB 4:4:4 to YCbCr 4:2:0), or performing a remapping of the input picture components in order to get a signal distribution more resilient to compression (for instance using a histogram equalization of one of the color components). Metadata may be associated with the pre-processing and attached to the bitstream.

[0096] In the encoder (200), a picture may be encoded by one or more encoder elements as described herein. The picture to be encoded may be partitioned (202) and processed in units of, for example, coding units (CUs). A unit may be encoded using, for example, an intra mode or an inter mode. If a unit is encoded in an intra mode, the encoder may perform intra prediction (260). If a unit is encoded in an inter mode, the encoder may perform motion estimation (275) and/or motion compensation (270). The encoder may decide (205) which one of the intra mode or inter mode to use for encoding the unit and may indicate the intra/inter decision by, for example, a prediction mode indication, such as a prediction mode flag. Prediction residuals may be calculated, for example, by subtracting (210) the predicted block from the image block, e.g., the original image block.

[0097] The prediction residuals may be transformed (225) and quantized (230). The quantized transform coefficients, as well as one or more motion vectors and/or other syntax elements, may be entropy coded (245) to output a bitstream. The encoder may skip the transform and apply quantization directly to the non-transformed residual signal. The encoder may bypass transform and/or quantization. For example, the residual may be coded directly without the application of the transform and/or the quantization processes.

[0098] The encoder may decode an encoded block to provide a reference for further predictions. The quantized transform coefficients may be de-quantized (240), and inverse transformed (250) to decode prediction residuals. Combining (255) the decoded prediction residuals and the predicted block, an image block may be reconstructed. In-loop filters (265) may be applied to the reconstructed picture to perform, for example, deblocking/sample adaptive offset (SAO) filtering to reduce encoding artifacts. The filtered image may be stored at a reference picture buffer (280).

[0099] FIG. 3 illustrates a diagram showing an example of a video decoder. In example decoder (300), a bitstream may be decoded by one or more decoder elements as described herein. The video decoder (300) may perform a decoding pass reciprocal to the encoding pass as described in FIG. 2. The encoder (200) may perform video decoding as part of encoding video data.

[0100] In examples, the input of the decoder may be, or may include, a video bitstream, which may be generated by a video encoder (200). The bitstream may be entropy decoded (330) to obtain transform coefficients, motion vectors, and/or other coded information. The picture partition information may indicate how the picture is partitioned. The decoder may divide (335) the picture according to the decoded picture partitioning information. The transform coefficients may be de-quantized (340), and inverse transformed (350) to decode the prediction residuals. Combining (355) the decoded prediction residuals and the predicted block, an image block may be reconstructed. The predicted block may be obtained (370) from intra prediction (360) or motion-compensated prediction (e.g., inter prediction) (375). In-loop filters (365) may be applied to the reconstructed image. The filtered image may be stored at a reference picture buffer (380). In examples, for a given picture, the contents of the reference picture buffer (380) on the decoder (300) side may be similar (e.g., identical) to the contents of the reference picture buffer (280) on the encoder (200) side for the same picture.

[0101] The decoded picture may go through post-decoding processing (385), for example, an inverse color transform (e.g., conversion from YCbCr 4:2:0 to RGB 4:4:4) and/or an inverse remapping performing the inverse of the remapping process performed in the pre-encoding processing (201). The post-decoding processing may use metadata derived in the pre-encoding processing and signaled in the bitstream. In an example, the decoded images (e.g., after application of the in-loop filters (365) and/or after post-decoding processing (385), if post-decoding processing is used) may be sent to a display device for rendering to a user.

[0102] FIG. 4 illustrates a diagram showing an example of a system in which various aspects and examples described herein may be implemented. A system (400) may be embodied as a device including the various components described below and may be configured to perform one or more of the aspects described in this document. Examples of such devices may include, but are not limited to, various electronic devices such as personal computers, laptop computers, smartphones, tablet computers, digital multimedia set top boxes, digital television receivers, personal video recording systems, connected home appliances, and servers. One or more elements of the system (400), singly or in combination, may be embodied in a single integrated circuit (IC), multiple ICs, and/or discrete components. For example, in at least one example, the processing and encoder/decoder elements of the system (400) may be distributed across multiple ICs and/or discrete components. In various examples, the system (400) may be communicatively coupled to one or more other systems, or other electronic devices, via, for example, a communications bus or through dedicated input and/or output ports. In various examples, the system (400) may be configured to implement one or more of the aspects described in this document.

[0103] The system (400) may include at least one processor (410) configured to execute instructions loaded therein for implementing, for example, the various aspects described herein. The processor (410) may include embedded memory, input output interface, and/or various other circuitries as known in the art. The system (400) may include at least one memory (420) (e.g., a volatile memory device, and/or a non-volatile memory device). The system (400) may include a storage device (440), which may include non-volatile memory and/or volatile memory, including, but not limited to, Electrically Erasable Programmable Read-Only Memory (EEPROM), Read-Only Memory (ROM), Programmable Read-Only Memory (PROM), Random Access Memory (RAM), Dynamic Random Access Memory (DRAM), Static Random Access Memory (SRAM), flash, magnetic disk drive, and/or optical disk drive. The storage device (440) may include an internal storage device, an attached storage device (including detachable and non-detachable storage devices), and/or a network accessible storage device, as non-limiting examples.

[0104] The system (400) may include an encoder/decoder module (430) configured, for example, to process data to provide an encoded video or decoded video, and the encoder/decoder module (430) may include its own processor and memory. The encoder/decoder module (430) may represent a module(s) that may be included in a device to perform the encoding and/or decoding functions. As is known, a device may include one or both of the encoding and decoding modules. Additionally, the encoder/decoder module (430) may be implemented as a separate element of system (400) or may be incorporated within the processor (410) as a combination of hardware and software as known to those skilled in the art.

[0105] Program code to be loaded onto the processor (410) or the encoder/decoder (430) to perform the various aspects described herein may be stored in the storage device (440) and subsequently loaded onto the memory (420) for execution by the processor (410). In accordance with various examples, one or more of the processor (410), the memory (420), the storage device (440), and the encoder/decoder module (430) may store one or more of various items during the performance of the processes described herein. Such stored items may include, but are not limited to, the input video, the decoded video or portions of the decoded video, the bitstream, matrices, variables, and intermediate or final results from the processing of equations, formulas, operations, and operational logic.

[0106] In some examples, a memory inside of the processor (410) and/or the encoder/decoder module (430) may be used to store instructions and to provide working memory for processing that is needed during encoding or decoding. In other examples, however, a memory external to the processing device (for example, the processing device may be either the processor (410) or the encoder/decoder module (430)) may be used for one or more of these functions. The external memory may be the memory (420) and/or the storage device (440), for example, a dynamic volatile memory and/or a non-volatile flash memory. In several examples, an external non-volatile flash memory is used to store the operating system of, for example, a television. In at least one example, a fast external dynamic volatile memory such as a RAM is used as working memory for video encoding and decoding operations.

[0107] The input to the elements of the system (400) may be provided through various input devices as indicated in the block (445). Such input devices may include, but are not limited to, (i) a radio frequency (RF) portion that receives an RF signal transmitted, for example, over the air by a broadcaster, (ii) a Component (COMP) input terminal (or a set of COMP input terminals), (iii) a Universal Serial Bus (USB) input terminal, and/or (iv) a High Definition Multimedia Interface (HDMI) input terminal. Other examples, not shown in FIG. 4, may include composite video.

[0108] In various examples, the input devices of the block (445) may have associated respective input processing elements as known in the art. For example, the RF portion may be associated with elements suitable for (i) selecting a desired frequency (also referred to as selecting a signal, or band-limiting a signal to a band of frequencies), (ii) downconverting the selected signal, (iii) band-limiting again to a narrower band of frequencies to select (for example) a signal frequency band which may be referred to as a channel in certain examples, (iv) demodulating the downconverted and band-limited signal, (v) performing error correction, and/or (vi) demultiplexing to select the desired stream of data packets. The RF portion of various examples may include one or more elements to perform these functions, for example, frequency selectors, signal selectors, band-limiters, channel selectors, filters, downconverters, demodulators, error correctors, and demultiplexers. The RF portion may include a tuner that performs various of these functions, including, for example, downconverting the received signal to a lower frequency (for example, an intermediate frequency or a near-baseband frequency) or to baseband. In one set-top box example, the RF portion and its associated input processing element may receive an RF signal transmitted over a wired (for example, cable) medium, and may perform frequency selection by filtering, downconverting, and filtering again to a desired frequency band. Various examples rearrange the order of the above-described (and other) elements, may remove some of these elements, and/or add other elements performing similar or different functions. Adding elements may include inserting elements in between existing elements, such as, for example, inserting amplifiers and an analog-to-digital converter. In various examples, the RF portion may include an antenna.

[0109] The USB and/or HDMI terminals may include respective interface processors for connecting system (400) to other electronic devices across USB and/or HDMI connections. It is to be understood that various aspects of input processing, for example, Reed-Solomon error correction, may be implemented, for example, within a separate input processing IC or within the processor (410) as necessary. Similarly, aspects of USB or HDMI interface processing may be implemented within separate interface ICs or within the processor (410) as necessary. The demodulated, error corrected, and demultiplexed stream may be provided to various processing elements, including, for example, the processor (410) and the encoder/decoder (430) operating in combination with the memory and the storage elements to process the datastream as necessary for presentation on an output device.

[0110] Various elements of system (400) may be provided within an integrated housing. Within the integrated housing, the various elements may be interconnected and transmit data therebetween using the suitable connection arrangement (425), for example, an internal bus as known in the art, including the Inter-IC (I2C) bus, wiring, and printed circuit boards.

[0111] The system (400) may include a communication interface (450) that enables communication with other devices via a communication channel (460). The communication interface (450) may include, but is not limited to, a transceiver configured to transmit and to receive data over the communication channel (460). The communication interface (450) may include, but is not limited to, a modem or network card and the communication channel (460) may be implemented, for example, within a wired and/or a wireless medium.

[0112] Data may be streamed, or otherwise provided, to the system (400), in various examples, using a wireless network such as a Wi-Fi network, for example IEEE 802.11 (IEEE refers to the Institute of Electrical and Electronics Engineers). The Wi-Fi signal of these examples is received over the communications channel (460) and the communications interface (450) which are adapted for Wi-Fi communications. The communications channel (460) of these examples may be typically connected to an access point or router that provides access to external networks including the Internet for allowing streaming applications and other over-the-top communications. Other examples may provide streamed data to the system (400) using a set-top box that delivers the data over the HDMI connection of the input block (445). Still other examples provide streamed data to the system (400) using the RF connection of the input block (445). As indicated above, various examples may provide data in a non-streaming manner. Additionally, various examples use wireless networks other than Wi-Fi, for example a cellular network or a Bluetooth® network.

[0113] The system (400) may provide an output signal to various output devices, including a display (475), speakers (485), and/or other peripheral devices (495). The display (475) of various examples may include one or more of, for example, a touchscreen display, an organic light-emitting diode (OLED) display, a curved display, and/or a foldable display. The display (475) may be for a television, a tablet, a laptop, a cell phone (mobile phone), or other device. The display (475) may also be integrated with other components (for example, as in a smart phone), or separate (for example, an external monitor for a laptop). The other peripheral devices (495) may include, in various examples, one or more of a stand-alone digital video disc (or digital versatile disc) (DVD, for both terms), a disk player, a stereo system, and/or a lighting system. Various examples may use one or more peripheral devices (495) that provide a function based on the output of the system (400). For example, a disk player performs the function of playing the output of the system (400).

[0114] In various examples, control signals are communicated between the system (400) and the display (475), the speakers (485), or other peripheral devices (495) using signaling such as AV.Link, Consumer Electronics Control (CEC), or other communications may protocol that enable device-to-device control with or without user intervention. The output devices may be communicatively coupled to system (400) via dedicated connections through respective interfaces (470, 480, and 490). Alternatively, the output devices may be connected to the system (400) using the communications channel (460) via the communications interface (450). The display (475) and speakers (485) may be integrated into a single unit with the other components of the system (400) in an electronic device such as, for example, a television. In various examples, the display interface (470) may include a display driver, such as, for example, a timing controller (T Con) chip.

[0115] The display (475) and the speakers (485) may alternatively be separate from one or more of the other components, for example, if the RF portion of input (445) is part of a separate set-top box. In various examples in which the display (475) and the speakers (485) may be external components, the output signal may be provided via dedicated output connections, including, for example, HDMI ports, USB ports, or COMP outputs.

[0116] The examples may be carried out by computer software implemented by the processor (410) or by hardware, or by a combination of hardware and software. As a non-limiting example, the examples may be implemented by one or more integrated circuits. The memory (420) may be of any type appropriate to the technical environment and may be implemented using any appropriate data storage technology, such as optical memory devices, magnetic memory devices, semiconductor-based memory devices, fixed memory, and removable memory, as non-limiting examples. The processor (410) may be of any type appropriate to the technical environment, and may encompass one or more microprocessors, general purpose computers, special purpose computers, and processors based on a multi-core architecture, as non-limiting examples.

[0117] Various implementations involve decoding. “Decoding,” as used in this application, can encompass all or part of the processes performed, for example, on a received encoded sequence in order to produce a final output suitable for display. In various examples, such processes include one or more of the processes typically performed by a decoder, for example, entropy decoding, inverse quantization, inverse transformation, and differential decoding. In various examples, such processes also, or alternatively, may include processes performed by a decoder of various implementations described in this application, for example, obtaining a prediction block of a current block in a current picture; obtaining a plurality of samples in the prediction block, wherein the plurality of samples include previously decoded samples and non-decoded samples; padding the non-decoded samples using neighboring samples; decoding the current block based on the padded samples, etc.

[0118] As further examples, in one example “decoding” refers only to entropy decoding, in another example “decoding” refers only to differential decoding, and in another example “decoding” refers to a combination of entropy decoding and differential decoding. Whether the phrase “decoding process” is intended to refer specifically to a subset of operations or generally to the broader decoding process will be clear based on the context of the specific descriptions and is believed to be well understood by those skilled in the art.

[0119] Various implementations involve encoding. In an analogous way to the above discussion about “decoding,” “encoding” as used in this application can encompass all or part of the processes performed, for example, on an input video sequence in order to produce an encoded bitstream. In various examples, such processes include one or more of the processes typically performed by an encoder, for example, partitioning, differential encoding, transformation, quantization, and entropy encoding. In various examples, such processes also, or alternatively, may include processes performed by an encoder of various implementations described in this application, for example, obtaining a prediction block of a current block in a current picture; obtaining a plurality of samples in the prediction block, wherein the plurality of samples include previously encoded samples and non-encoded samples; padding the non- encoded samples using neighboring samples; encoding the current block based on the padded samples, etc.

[0120] As further examples, in one example “encoding” refers only to entropy encoding, in another example “encoding” refers only to differential encoding, and in another example “encoding” refers to a combination of differential encoding and entropy encoding. Whether the phrase “encoding process” is intended to refer specifically to a subset of operations or generally to the broader encoding process will be clear based on the context of the specific descriptions and is believed to be well understood by those skilled in the art.

[0121] When a figure is presented as a flow diagram, it should be understood that it also provides a block diagram of a corresponding apparatus. Similarly, when a figure is presented as a block diagram, it should be understood that it also provides a flow diagram of a corresponding method/process.

[0122] The implementations and aspects described herein may be implemented in, for example, a method or a process, an apparatus, a software program, a data stream, or a signal. Even if only discussed in the context of a single form of implementation (for example, discussed only as a method), the implementation of features discussed can also be implemented in other forms (for example, an apparatus or program). An apparatus may be implemented in, for example, appropriate hardware, software, and firmware. The methods may be implemented in, for example, a processor, which refers to processing devices in general, including, for example, a computer, a microprocessor, an integrated circuit, or a programmable logic device. Processors also include communication devices, such as, for example, computers, cell phones, portable/personal digital assistants (PDAs), and other devices that facilitate communication of information between end-users.

[0123] Reference to “one example” or “an example” or “one implementation” or “an implementation”, as well as other variations thereof, means that a particular feature, structure, characteristic, and so forth described in connection with the example is included in at least one example. Thus, the appearances of the phrase “in one example” or “in an example” or “in one implementation” or “in an implementation”, as well any other variations, appearing in various places throughout this application are not necessarily all referring to the same example.

[0124] Additionally, this application may refer to “determining” various pieces of information. Determining the information can include one or more of, for example, estimating the information, calculating the information, predicting the information, or retrieving the information from memory. Obtaining may include receiving, retrieving, constructing, generating, and/or determining.

[0125] Further, this application may refer to “accessing” various pieces of information. Accessing the information can include one or more of, for example, receiving the information, retrieving the information (for example, from memory), storing the information, moving the information, copying the information, calculating the information, determining the information, predicting the information, or estimating the information.

[0126] Additionally, this application may refer to “receiving” various pieces of information. Receiving is, as with “accessing”, intended to be a broad term. Receiving the information can include one or more of, for example, accessing the information, or retrieving the information (for example, from memory). Further, “receiving” is typically involved, in one way or another, during operations such as, for example, storing the information, processing the information, transmitting the information, moving the information, copying the information, erasing the information, calculating the information, determining the information, predicting the information, or estimating the information.

[0127] It is to be appreciated that the use of any of the following “/”, “and/or”, and “at least one of”, for example, in the cases of “A/B”, “A and/or B” and “at least one of A and B”, is intended to encompass the selection of the first listed option (A) only, or the selection of the second listed option (B) only, or the selection of both options (A and B). As a further example, in the cases of “A, B, and/or C” and “at least one of A, B, and C”, such phrasing is intended to encompass the selection of the first listed option (A) only, or the selection of the second listed option (B) only, or the selection of the third listed option (C) only, or the selection of the first and the second listed options (A and B) only, or the selection of the first and third listed options (A and C) only, or the selection of the second and third listed options (B and C) only, or the selection of all three options (A and B and C). This may be extended, as is clear to one of ordinary skill in this and related arts, for as many items as are listed.

[0128] Also, as used herein, the word “signal” refers to, among other things, indicating something to a corresponding decoder. Encoder signals may include, for example, an encoding function on an input for a block using a precision factor, etc. In this way, in an example, the same parameter is used at both the encoder side and the decoder side. Thus, for example, an encoder can transmit (explicit signaling) a particular parameter to the decoder so that the decoder can use the same particular parameter. Conversely, if the decoder already has the particular parameter as well as others, then signaling may be used without transmitting (implicit signaling) to simply allow the decoder to know and select the particular parameter. By avoiding transmission of any actual functions, a bit savings may be realized in various examples. It is to be appreciated that signaling may be accomplished in a variety of ways. For example, one or more syntax elements, flags, and so forth are used to signal information to a corresponding decoder in various examples. While the preceding relates to the verb form of the word “signal,” the word “signal” may (e.g., may also) be used herein as a noun.

[0129] As will be evident to one of ordinary skill in the art, implementations may produce a variety of signals formatted to carry information that may be, for example, stored or transmitted. The information can include, for example, instructions for performing a method, or data produced by one of the described implementations. For example, a signal may be formatted to carry the bitstream of a described example. Such a signal may be formatted, for example, as an electromagnetic wave (for example, using a radio frequency portion of the spectrum) or as a baseband signal. The formatting may include, for example, encoding a data stream and modulating a carrier with the encoded data stream. The information that the signal carries may be, for example, analog or digital information. The signal may be transmitted over a variety of different wired or wireless links, as is known. The signal may be stored on, or accessed, or received from a processor-readable medium.

[0130] Many examples are described herein. Features of examples may be provided alone or in any combination, across various claim categories and types. Further, examples may include one or more of the features, devices, or aspects described herein, alone or in any combination, across various claim categories and types. For example, features described herein may be implemented in a bitstream or signal that includes information generated as described herein. The information may allow a decoder to decode a bitstream, the encoder, bitstream, and/or decoder according to any of the embodiments described. For example, features described herein may be implemented by creating and/or transmitting and/or receiving and/or decoding a bitstream or signal. For example, features described herein may be implemented a method, process, apparatus, medium storing instructions, medium storing data, or signal. For example, features described herein may be implemented by a TV, set-top box, cell phone, tablet, or other electronic device that performs decoding. The TV, set-top box, cell phone, tablet, or other electronic device may display (e.g., using a monitor, screen, or other type of display) a resulting image (e.g., an image from residual reconstruction of the video bitstream). The TV, set-top box, cell phone, tablet, or other electronic device may receive a signal including an encoded image and perform decoding.

[0131] These examples may be performed by a device with at least one processor. The device may be an encoder or a decoder. These examples may be performed by a computer program product which is stored on a non-transitory computer readable medium and includes program code instructions. These examples may be performed by a computer program comprising program code instructions.

[0132] Intra block copy (IBC) may be used for screen content coding. IBC may improve the coding efficiency of screen content materials. IBC mode may be implemented as a block level coding mode. For example, because the IBC mode is implemented as a block level coding mode, block matching (BM) may be performed at an encoder, e.g., to find a block vector (e.g., and/or motion vector) for a coding unit (CU). A block vector may indicate a displacement from a current block to a reference block. The reference block may be (e.g., may already be) reconstructed inside the current picture. The luma block vector of an IBC-coded CU may be in an integer precision. The chroma block vector may be rounded to the integer precision. If combined with adaptive motion vector resolution (AMVR), the IBC mode may switch between 1-pel and 4-pel motion vector precisions. An IBC-coded CU may be treated as the third prediction mode, e.g., other than intra or inter prediction modes. The IBC mode may be applicable to one or more CUs with a width and/or a height smaller than or equal to 64 luma samples.

[0133] At a CU level, the IBC mode may be signaled with an indication, such as a flag. The IBC mode may be signaled as an IBC advanced motion vector prediction (AMVP) mode or an IBC skip/merge mode. In examples, an IBC skip/merge mode may use a merge candidate index. For example, a merge candidate index may be used to indicate which of the block vectors in the list from neighboring candidate IBC coded blocks is used to predict the current block. The merge list may include spatial, history-based motion vector prediction (HMVP), and/or pairwise candidates. In examples, an IBC AMVP mode may use block vector difference. For example, block vector difference may be coded in the same way as a motion vector difference. The block vector prediction may use two candidates as predictors, e.g., one from a left neighbor and the other from an above neighbor (e.g., if IBC coded). If either neighbor is not available, a block vector (e.g., a default block vector) may be used as a predictor. An indication, such as a flag, may be signaled to indicate the block vector predictor index.

[0134] An IBC reference region may be utilized. For example, an IBC reference region may be utilized to limit memory consumption and/or decoder complexity. The IBC in video coding may allow the reconstructed portion of the predefined area, e.g., including the region of current CTU and some region of the left CTU.

[0135] FIGs. 5A-5D illustrate examples of IBC reference regions depending on a current block prediction. For example, FIGs. 5A-5D may illustrate current coding tree unit (CTU) processing order and available reference samples in current and left CTU. FIGs. 5A-5D illustrate the reference regions of IBC mode, where a block may represent 64x64 luma sample unit.

[0136] Depending on the location of the current coding CU location within the current CTU, one or more of the following may apply.

[0137] In examples, if a current block falls into the top-left 64x64 block of the current CTU, in addition to the reconstructed samples (e.g., already reconstructed samples) in the current CTU, the current block may also refer to the reference samples in the bottom-right 64x64 blocks of the left CTU, e.g., using a current picture referencing (CPR) mode. The current block may also refer to the reference samples in the bottom-left 64x64 block of the left CTU and the reference samples in the top-right 64x64 block of the left CTU, e.g., using CPR mode.

[0138] In examples, if the current block falls into the top-right 64x64 block of the current CTU, in addition to the already reconstructed samples in the current CTU, if luma location (0, 64) relative to the current CTU has not yet been reconstructed, the current block may also refer to the reference samples in the bottom-left 64x64 block and bottom-right 64x64 block of the left CTU, e.g., using CPR mode. If the luma location (0, 64) relative to the current CTU has been reconstructed, the current block may also refer to reference samples in bottom-right 64x64 block of the left CTU.

[0139] In examples, if the current block falls into the bottom-left 64x64 block of the current CTU, in addition to the already reconstructed samples in the current CTU, if luma location (64, 0) relative to the current CTU has not yet been reconstructed, the current block may also refer to the reference samples in the top-right 64x64 block and bottom-right 64x64 block of the left CTU, e.g., using CPR mode. If the luma location (0, 64) relative to the current CTU has been reconstructed, the current block may also refer to the reference samples in the bottom-right 64x64 block of the left CTU, e.g., using CPR mode.

[0140] In examples, if the current block falls into the bottom-right 64x64 block of the current CTU, the current block may refer to the already reconstructed samples in the current CTU, e.g., using CPR mode.

[0141] The location of the current coding CU location within the current CTU described herein may allow the IBC mode to be implemented, e.g., using a local on-chip memory for hardware implementations.

[0142] The IBC merge/AMVP list construction may be modified. In examples, if an IBC merge/AMVP candidate is valid, the IBC merge/AMVP candidate may be inserted into the IBC merge/AMVP candidate list. In examples, above-right, bottom-left, and/or above-left spatial candidates and a pairwise average candidate may be added into the IBC merge/AMVP candidate list. In examples, a template based adaptive reordering (ARMC-TM) may be applied to the IBC merge list.

[0143] The history-based motion vector prediction (HMVP) table size for an IBC may be increased, e.g., to 25 entries. After up to 20 IBC merge candidates are derived with full pruning, the IBC merge candidates may be reordered together. After reordering, one or more candidates may be selected as final candidates in the IBC merge list. For example, after reordering, the first 6 candidates with the lowest template matching costs may be selected as the final candidates in the IBC merge list.

[0144] One or more candidates for zero vectors to pad the IBC Merge/AMVP list may be replaced with a set of block vector prediction (BVP) candidates located in the IBC reference region. A zero vector may be invalid as a block vector in the IBC merge mode, and the zero vector may be discarded as the BVP in the IBC candidate list.

[0145] FIG. 6 illustrates one or more padding candidates for the replacement of a zero-vector in an IBC list. For example, as illustrated in FIG. 6, two or more (e.g., three) candidates may be located on corners (e.g., the nearest corners) of the reference region. Two or more (e.g., three) additional candidates may be determined in the middle of the three sub-regions (A, B, and C). The coordinates may be determined by the width and height of the current block and/or the ΔX and ΔY parameters, as illustrated in Fig. 6.

[0146] In examples, the reference for IBC may be extended to two CTU rows above the CTU being processed by the encoder or the encoder. FIG. 7 illustrates an example reference area for IBC if a CTU (m,n) is coded. As illustrated in FIG. 7, for CTU (m,n) to be coded, the reference area may include CTUs with index (m–2,n–2)…(W,n–2),(0,n–1)…(W,n–1),(0,n)…(m,n), where W denotes the maximum horizontal index within the current tile, slice, or picture. The per-sample block vector search (e.g., or called local search) range may be limited to $[-(C \ll 1), C \gg 2]$ horizontally and $[-C, C \gg 2]$ vertically to adapt to the reference area extension, where C denotes the CTU size.

[0147] In examples, template matching (TM) based motion search and refinement may be applied to the case of IBC. For example, an IBC-TM merge mode may be used. The IBC-TM may involve a merge candidate list for block vector (BV) prediction, e.g., different from the one used by regular IBC merge mode. The one or more candidates may be selected according to a pruning method with a motion distance between the candidates as in the regular TM merge mode. The zero motion candidates may have been replaced by (-W, 0), (0, -H), (-W, -H) motion vectors (MVs).

[0148] In the IBC-TM merge mode, the selected candidates may be refined with the template matching method. A TM-merge indication, such as a TM-merge flag, may be signaled to indicate the template matching merge IBC mode.

[0149] In the IBC-TM AMVP mode, up to 3 candidates may be selected from the IBC-TM merge list. A candidate (e.g., each of the candidates) may be refined according to the template matching method and/or may be sorted according to the resulting TM cost.

[0150] If the IBC is used, TM refinement may be performed, e.g., at integer pel position, and in IBC-TM AMVP mode, TM refinement may be performed at integer or 4-pel precision, e.g., depending on the AMVR value. The refinement may be done within an area, e.g., the existing IBC reference area.

[0151] IBC mode may interact with one or more coding tools. For example, the interaction between IBC mode and one or more other inter coding tools, such as pairwise merge candidate, history-based motion vector predictor (HMVP), combined intra/inter prediction mode (CIIP), merge mode with motion vector difference (MMVD), and/or geometric partitioning mode (GPM) may be as follows.

[0152] In examples, IBC may be used with pairwise merge candidate and HMVP. A pairwise IBC merge candidate (e.g., a new pairwise IBC merge candidate) may be generated by averaging two IBC merge candidates. For HMVP, IBC motion may be inserted into history buffer for future referencing.

[0153] In examples, IBC may not be used in combination with an inner tool, such as affine motion.

[0154] In examples, IBC may be used in combination with CIIP, MMVD, and GPM.

[0155] In examples, IBC may not be allowed for the chroma coding blocks if a partition, such as DUAL_TREE partition, is used.

[0156] In a coding tool, the current picture may not be included as one of the reference pictures in the reference picture list 0 for IBC prediction. The derivation process of motion vectors for IBC mode may exclude one or more (e.g., all) neighboring blocks in inter mode and vice versa. The following IBC design aspects may be applied in a coding tool.

[0157] In examples, IBC may share the same process as in regular MV merge including with pairwise merge candidate and history-based motion predictor but disallows TMVP and zero vector because TMVP and zero vector may be invalid for IBC mode.

[0158] In examples, separate HMVP buffer (e.g., 5 candidates each) may be used for MV (e.g., conventional MV) and IBC.

[0159] In examples, block vector constraints may be implemented in the form of bitstream conformance constraint. The encoder may ensure that no invalid vectors are present in the bitstream, and merge may not be used if the merge candidate is invalid (out of range or 0). Such bitstream conformance constraint may be expressed in terms of a virtual buffer as described herein.

[0160] In example, for deblocking, IBC may be handled as inter mode.

[0161] In examples, if the current block is coded using IBC prediction mode, AMVR may not use quarter-pel. AMVR may be signaled to indicate whether MV is inter-pel or 4 integer-pel.

[0162] In example, the number of IBC merge candidates may be signaled in the slice header separately from the numbers of regular, sub-block, and/or geometric merge candidates.

[0163] Merge with Moton vector Difference (MMVD) used in a coding tool (e.g., associated with inter-predicted blocks) may include one or more of the following.

[0164] In examples, affine-MMVD and GPM-MMVD may be an extension of regular MMVD mode. The MMVD mode may be extended to the IBC merge mode.

[0165] In examples, in the IBC-block vector difference (IBC-MBVD), a motion vector difference the distance set may be {1-pel, 2-pel, 4-pel, 8-pel, 12-pel, 16-pel, 24-pel, 32-pel, 40-pel, 48-pel, 56-pel, 64-pel, 72-pel, 80-pel, 88-pel, 96-pel, 104-pel, 112-pel, 120-pel, 128-pel}, and the BVD directions may be two horizontal and two vertical directions.

[0166] In examples, the base candidates may be selected from the candidates (e.g., the first five candidates) in the reordered IBC merge list. Based on the sum of absolute differences (SAD) cost between the template (e.g., one row above and one column left to the current block) and the reference for a refinement position, one or more (e.g., all) the possible MBVD refinement positions (e.g., 20×4) for the base candidate may be reordered. The top 8 refinement positions with the lowest template SAD costs may be kept as available positions, consequently for MBVD index coding. The MBVD index may be binarized by the rice code with the parameter equal to 1.

[0167] IBC may be adapted to camera-captured video content. In a coding tool, such as EMC, the IBC coding mode may be used by default for the coding of camera-captured content. One or more adaptations, such as high-level tool control, encoder optimization, and/or fraction-pel extension on IBC, may be adopted to make compression efficient.

[0168] High level tool control may be adopted to make compression efficient.

[0169] In examples, IBC merge modes may be disabled for natural content. A sequence parameter set (SPS) level indication, such as an SPS level flag, may be implemented to disable the associated CU level signaling. IBC AMVP modes may be activated if the IBC AMVP modes are indicated explicitly by the SPS indication, such as the SPS flag.

[0170] In examples, reconstruction-reordered IBC (RR-IBC), TM-IBC, and/or IBC-CIIP may be disabled for natural content. For RR-IBC and TM-IBC, corresponding SPS indications (e.g., SPS flags) may be implemented.

[0171] IBC may be applied to intra slices for natural content. For example, IBC being applied to intra slices for natural content may be indicated by a high-level syntax to remove CU level signaling of an IBC indication (e.g., an IBC flag). For screen contents, IBC may be applied to one or more (e.g., all) slices.

[0172] Encoder optimization may be adopted to make compression efficient.

[0173] In examples, at the encoder, IBC block vector search may be optimized for natural content, and the rate-distortion optimization (RDO) process may be skipped for an IBC AMVP mode if the SAD cost is much worse than the lowest SAD cost of one or more (e.g., all) Intra modes.

[0174] In examples, IBC AMVP modes may not be evaluated if the best Intra mode has less than 3 nonzero coefficients.

[0175] In a coding tool, some partitioning depth in an inter slice may be skipped, e.g., depending on the picture order count (POC) distance between the current picture and the nearest reference picture of the current picture. If IBC is enabled from the SPS level, a coding tool may set the POC distance equal to 0, which may not align with the configuration used in common test conditions (CTC) for Random Access and Low Delay. In the test, for inter slices, the true POC distance may be used, e.g., instead of setting to 0.

[0176] Fraction-pel extension on IBC may be adopted to make compression efficient. The representation of IBC block vectors may be extended to fractional-pel resolution. An interpolation filter may be needed to derive the prediction samples located at a non-integer phase in the reconstructed area of the current frame.

[0177] In examples, the option of block vector resolutions may be extended to include quarter-pel resolution in additional to full-pel and 4-pel ones. Similar to inter AMVR syntax, the first bin of AMVR syntax may be signaled to indicate whether BV is in quarter-pel resolution. The second bin may be signaled to switch between full-pel and 4-pel resolutions.

[0178] In examples, the interpolation filters applied to the luma and chroma components of IBC blocks may be the 8-tap luma filter and the chroma filter as used in motion compensation, respectively. A 2-tap bilinear interpolation filter may be applied to generating template prediction blocks, if needed the IBC-related coding tools.

[0179] In examples, reference sample padding may be needed if one or more reference samples are not available or located outside a valid IBC reference area in the current frame. If needed, the reference sample padding may perform in horizontal direction first and then vertical direction.

[0180] Bi-predictive IBC prediction mode may be used. The bi-predictive IBC may have two prediction modes: IBC BVP-merge mode and bi-predictive IBC merge mode.

[0181] IBC BVP-merge mode may derive the two BVs from IBC BVP mode and IBC merge mode, e.g., similar to the MV derivation of AMVP-merge mode which combines an AMVP motion vector predictor for a reference list and an inter merge candidate for the other reference list, to form a bi-predicted inter CU. Two different indices for the IBC BVP mode and the IBC merge candidate may be signaled from the encoder to the decoder, respectively taken from IBC AMVP candidate list and IBC merge candidate list.

[0182] Bi-predictive IBC merge mode may derive the two BVs from the IBC merge candidate list, e.g., utilizing two different IBC merge indices. The two indices may be signaled from the encoder to the decoder. The target of the bi-predictive IBC merge mode may be IBC-regular merge and/or IBC merge mode with block vector difference (IBC-MBVD) and IBC geometric partitioning mode (IBC-GPM) (e.g., which may be enabled in a coding tool, for screen content by default). In bi-predictive IBC merge mode, bi-predictive IBC-MBVD may be enabled in natural and screen content, and bi-predictive IBC-GPM may be enabled in screen content.

[0183] IBC BVP-merge mode and bi-predictive IBC merge mode may use one or more of the following: merge candidate list construction; BV refinement; compensation; BV storage; signaling; and/or enabling in chroma component blocks.

[0184] Merge candidate list may be constructed. IBC BVP-merge mode and bi-predictive IBC merge mode may reuse the IBC merge candidate list construction method (e.g., existing IBC merge candidate list construction method) for uni-predictive IBC merge mode.

[0185] BV may be refined. IBC BVP-merge mode and bi-predictive IBC merge mode may enable the IBC (e.g., the existing IBC) with template matching.

[0186] IBC BVP-merge mode and bi-predictive IBC merge mode may generate final IBC prediction samples with a (1:1) average of bi-predictive IBC samples.

[0187] Two BVs may be stored in BV storage if the bi-predictive IBC is enabled.

[0188] A control indication, such as a control flag, of bi-predictive IBC may be signaled at a slice level in I slice. The control indication, such as the control flag, of bi-predictive IBC may skip signaling in B and/or P slices. Reconstructed-Reordered IBC may be disabled if the bi-predictive IBC is enabled.

[0189] IBC BVP-merge mode and bi-predictive IBC merge mode may be enabled in chroma component blocks of a tree (e.g., the single tree).

[0190] In a coding tool, the motion data representation may be divided into two categories: whole-block-based motion representation and sub-block-based motion representation. FIG. 8 illustrates an example of whole-block and sub-block-based motion representation categories. As illustrated in FIG, 8, in the whole-block category and/or the sub-block-based motion representation category, merge/skip and AMVP modes for coding the motion information may be used.

[0191] The whole-block-based motion representation may be or may include in assigning a set of motion information, made of one or two motion vectors and associating a reference picture(s) to an inter block. For the whole-block-based motion representation, a motion vector may be used to describe the motion of the motion before refinement. For the sub-block motion representation, the motion may be described for sub-block.

[0192] In examples, the sub-block-based motion coding mode may divide a block into 4x4 or 8x8 luma samples sub-blocks and assign a set (e.g., an individual set) of motion information to a sub-block.

[0193] Sub-block-based motion representation and coding may be associated with one or more of the following: Affine motion compensation, Affine Merge mode, and/or Affine AMVP mode.

[0194] Sub-block-based motion representation and coding may be associated with Affine motion compensation. In examples, in a coding tool, a translation motion model may be applied for motion compensated temporal prediction (MCP). Such translational motion may be unable to capture one or more types of motion, such as zoom in, zoom out, rotation, perspective motions, and/or irregular motions. In examples, in other coding tool, a sub-block-based affine motion compensation prediction may be used at a CU level. FIGs. 9A-9B illustrate an example control point based affine motion models. FIG. 9A illustrates an example of 4-parameter affine model. Fig. 9B illustrates an example of 6-parameter affine model. The affine motion field of the block may be described by motion information of two control point motion vectors (e.g., 4-parameter affine motion model as illustrated in FIG. 9A) or three control point motion vectors (e.g., 6-parameter affine motion model as illustrated in FIG. 9B). As illustrated in FIGs. 9A-9B, he vectors v_0, v_1, v_2, may be the control point motion vectors (CPMVs) associated to the block and used to represent the affine motion field of the considered block.

[0195] For the 4-parameter affine motion model, motion vector at sample location (x, y) in a block may be derived as:

$$\begin{cases} mv_x = \frac{mv_{1x} - mv_{0x}}{W}x + \frac{mv_{0y} - mv_{1y}}{W}y + mv_{0x} \\ mv_y = \frac{mv_{1y} - mv_{0y}}{W}x + \frac{mv_{1x} - mv_{0x}}{W}y + mv_{0y} \end{cases}$$

Equation 1: 4-parameter affine motion field computation.

[0196] For the 6-parameter affine motion model, motion vector at sample location (x, y) in a block may be derived as:

$$\begin{cases} mv_x = \frac{mv_{1x} - mv_{0x}}{W}x + \frac{mv_{2x} - mv_{0x}}{H}y + mv_{0x} \\ mv_y = \frac{mv_{1y} - mv_{0y}}{W}x + \frac{mv_{2y} - mv_{0y}}{H}y + mv_{0y} \end{cases}$$

Equation 2: 6-parameter affine motion field computation.

[0197] Motion vector (mv0x, mv0y) may be the motion vector of the top-left corner control point. (mv1x, mv1y) may be the motion vector of the top-right corner control point. Motion vector (mv2x, mv2y) may be the motion vector of the bottom-left corner control point.

[0198] In a coding tool, affine motion compensation may be performed on a 4x4 sub-block basis. To derive motion vector of a 4×4 luma sub-block, the motion vector of the center sample of the sub-block according to above equations, as illustrated in FIGs. 9A-9B, may be calculated according to above equations and rounded to 1/16 fraction accuracy. The motion compensation interpolation filters may be applied to generate the prediction of a sub-block with derived motion vector. The sub-block size in chroma-components may be 4×4. The MV of a 4×4 chroma sub-block may be calculated as the average of the MVs of the top-left and bottom-right luma sub-blocks in the collocated 8x8 luma region.

[0199] FIG. 10 illustrates an example affine motion field representation on a 4x4 sub-block basis. As for translational motion inter prediction, affine AMVP mode and affine merge mode may be used for affine inter prediction modes.

[0200] In examples, Affine Merge mode may be used for affine inter prediction mode. The affine merge mode may be a sub-block-based motion coding mode inside the sub-block merge mode. The affine merge mode may be applied for one or more CUs with a width and/or a height larger than or equal to 8. In the affine merge mode, the CPMVs of the current CU may be generated based on the motion information of one or more spatial neighboring CUs. There may be up to five control point motion vector predictors (CPMVPs) candidates and an index may be signaled to indicate the one to be used for the current CU. One of the following types of CPVM candidates may be used to form the affine merge candidate list: inherited affine merge candidates extrapolated from the CPMVs of the neighbor CUs; constructed affine merge candidates CPMVPs that are derived using the translational MVs of the neighbor CUs; or zero MVs.

[0201] In a coding tool, one or more (e.g., maximum two) inherited affine candidates may be derived from affine motion model of the neighboring blocks, e.g., one from left neighboring CUs and one from above neighboring CUs.

[0202] FIG. 11 illustrates example locations of inherited affine motion predictors. The candidate blocks may be shown in FIG. 11. For the left predictor, the scan order may be from A0 to A1. As illustrated in FIG. 11, for the above predictor, the scan order may be from B0 to B1 to B2. The first inherited candidate from each side may be selected. If a neighboring affine CU is identified, one or more control point motion vectors of the neighboring affine CU may be used to derive the CPMVP candidate in the affine merge list of the current CU.

[0203] FIG. 12 illustrates an example CPMV inheritance from an affine block to another affine block. As illustrated in FIG. 12, if the neighbor left bottom block A is coded in affine mode, the motion vectors v_2, v_3, and v_4 of the top left corner, above right corner and left bottom corner of the CU, which contains the block A, may be attained. If block A is coded with 4-parameter affine model, the two CPMVs of the current CU may be calculated according to v_2 and v_3. If block A is coded with 6-parameter affine model, the three CPMVs of the current CU may be calculated according to v_2, v_3, and v_4.

[0204] Constructed affine candidate may be or may mean the candidate is constructed by combining the neighbor translational motion information of each control point. The motion information for the control points may be derived from the specified spatial neighbors and temporal neighbor illustrated in FIG. 13. CPMVk (k=1, 2, 3, 4) may represent the k-th control point. For CPMV1, the B2->B3->A2 blocks may be checked and the MV of the first available block may be used. For CPMV2, the B1->B0 blocks may be checked. For CPMV3, the A1->A0 blocks may be checked. TMVP may be used as CPMV4 if TMVP is available.

[0205] After MVs of four control points are attained, affine merge candidates may be constructed based on the motion information. The following combinations of control point MVs may be used to generate constructed affine merge candidate:

{CPMV1, CPMV2, CPMV3}, {CPMV1, CPMV2, CPMV4}, {CPMV1, CPMV3, CPMV4},
{CPMV2, CPMV3, CPMV4}, {CPMV1, CPMV2}, {CPMV1, CPMV3}.

[0206] For example, if control point motion vectors {CPMV1, CPMV2, CPMV3} are used, the control point motion vectors {CPMV1, CPMV2, CPMV3} may be used to generate an affine motion field for the CU, e.g., following FIG. 13.

[0207] The combination of 3 CPMVs as above constructs a 6-parameter affine merge candidate and the combination of 2 CPMVs may construct a 4-parameter affine merge candidate. To avoid motion scaling process in case CPMVs point to different reference pictures, if the reference indices of control points are different, the related combination of control point MVs may be discarded.

[0208] FIG. 13 illustrates example locations of candidate positions for constructed affine merge mode. After inherited affine merge candidates and constructed affine merge candidate are considered for being appended to the affine merge candidate list, if the list is still not full, one or more zero MVs may be inserted to the end of the list.

[0209] In examples, Affine AMVP mode may be used for affine inter prediction mode. Affine AMVP mode may be applied for one or more CUs with a width and/or a height larger than or equal to 16. An affine indication, such as an affine flag, at a CU level may be signaled, e.g., in a bitstream, to indicate the use of affine AMVP mode. Another indication, such as another flag, may be signaled if 4-parameter affine or 6-parameter affine model is used. In affine AMVP mode, the difference of the CPMVs of current CU and the predictors Control Point Motion Vector Predictors (CPMVPs) may be coded.

[0210] The CPMVPs used to predict the CPMV of a CU may be taken from an affine AMVP candidate list, e.g., made of two elements. The affine AMVP candidate list may be constructed using one or more of the following types of CPVM candidate (e.g., in order): inherited affine AMVP candidates extrapolated from the CPMVs of the neighbor CUs; constructed affine AMVP candidates CPMVPs that are derived using the translational MVs of the neighbor CUs; translational MVs from neighboring CUs; and/or Zero MVs.

[0211] One or more checking described herein may be configured. Checking a potential candidate may be or may mean checking that a valid Affine AMVP or Affine merge candidate to predict the current CU's affine CPMVs is available and is valid, and if so, add it to the candidate list under construction. The checking order of inherited affine AMVP candidates may be the same as the checking order of inherited affine merge candidates. The difference may be that, for AVMP candidate, the affine CU that has the same reference picture as in current block may be considered. Pruning process may be skipped if inserting an inherited affine motion predictor into the candidate list.

[0212] Constructed affine AMVP candidate may be derived from one or more spatial neighbors (e.g., one or more specified spatial neighbors) illustrated in FIG. 13. The same checking order may be used as in affine merge candidate construction. In addition to and/or alternatively, a reference picture index of the neighboring block may be checked. The first block in the checking order that is inter coded and has the same reference picture as in current CUs may be used.

[0213] If the current CU is coded with 4-parameter affine mode and mv_0 and mv_1 are available, the MVs may be added as a candidate in the affine AMVP list. If the current CU is coded with 6-parameter affine mode, and one or more (e.g., all) three CPMVs are available, the CPMVs may be added as one candidate in the affine AMVP list. If the current CU is not coded with 4-parameter affine mode and/or if mv_0 and mv_1 are unavailable and/or if the current CU is not coded with 6-parameter affine mode and/or if one or more (e.g., all) three CPMVs are unavailable, constructed AMVP candidate may be set as unavailable.

[0214] If affine AMVP list of candidates is still less than 2 after valid inherited affine AMVP candidates and constructed AMVP candidate are inserted, mv_0, mv_1, and mv_2 may be added, e.g., in order, as the translational MVs to predict one or more (e.g., all) control point MVs of the current CU, if available. Zero MVs may be used to fill the affine AMVP list if the list is not full.

[0215] Affine motion compensation refinement with prediction refinement with optical flow (PROF) may be configured. Sub-block based affine motion compensation may save memory access bandwidth and/or reduce computation complexity compared to pixel-based motion compensation, e.g., at the cost of prediction accuracy penalty. To achieve a finer granularity of motion compensation, PROF may be used to refine the sub-block based affine motion compensated prediction without increasing the memory access bandwidth for motion compensation. In a coding tool, after the sub-block based affine motion compensation is performed, luma prediction sample may be refined by adding a difference derived by the optical flow equation. The PROF may be described in the following: performing the sub-block-based affine motion compensation; calculating the spatial gradients $g_x(i,j)$ and $g_y(i,j)$ of the sub-block prediction; calculating the luma prediction refinement; and/or adding the luma prediction refinement $\Delta I(i,j)$ to the sub-block prediction $I(i,j)$.

[0216] The sub-block-based affine motion compensation may be performed to generate sub-block prediction $I(i,j)$.

[0217] The spatial gradients $g_x(i,j)$ and $g_y(i,j)$ of the sub-block prediction may be calculated at a sample location, e.g., using a 3-tap filter [−1, 0, 1]. The gradient calculation may be the same as follows.

$$g_x(i,j) = (I(i+1,j) \gg shift1) - (I(i-1,j) \gg shift1)$$

$$g_y(i,j) = (I(i,j+1) \gg shift1) - (I(i,j-1) \gg shift1).$$

[0218] $shift1$ may be used to control the gradient's precision. The sub-block (e.g., 4x4) prediction may be extended by a sample on a side for the gradient calculation. To avoid additional memory bandwidth and/or additional interpolation computation, the extended samples on the extended borders may be copied from the nearest integer pixel position in the reference picture.

[0219] The luma prediction refinement may be calculated by the following optical flow equation.

$$\Delta I(i,j) = g_x(i,j) * \Delta v_x(i,j) + g_y(i,j) * \Delta v_y(i,j),$$

where the $\Delta v(i,j)$ may be the difference between sample MV computed for sample location (i,j), denoted by $v(i,j)$. The sub-block MV of the sub-block to which sample (i,j) belongs, as illustrated in FIG. 14. The $\Delta v(i,j)$ may be quantized in the unit of 1/32 luma sample precision.

[0220] FIG. 14 illustrates an example subblock MV V_{SB} and pixel $\Delta v(i,j)$ (e.g., illustrated as the dotted arrow).

[0221] Since the affine model parameters and the sample location relative to the sub-block center are unchanged from sub-block to sub-block, $\Delta v(i,j)$ may be calculated for the first sub-block and reused for other sub-blocks in the same CU. For example, let $dx(i,j)$ and $dy(i,j)$ be the horizontal and vertical offset from the sample location (i,j) to the center of the sub-block (x_{SB}, y_{SB}). $\Delta v(x,y)$ may be derived by the following equation:

$$\begin{cases} \mathrm{dx}(i,j) = i - x_{SB} \\ \mathrm{dy}(i,j) = j - y_{SB} \end{cases}$$

$$\begin{cases} \Delta v_x(i,j) = C * dx(i,j) + D * dy(i,j) \\ \Delta v_y(i,j) = E * dx(i,j) + F * dy(i,j) \end{cases}$$

[0222] In order to keep accuracy, the center of the sub-block (x_{SB}, y_{SB}) may be calculated as ((WSB − 1)/2, (HSB − 1) / 2), where WSB and HSB are the sub-block width and height, respectively.

[0223] For 4-parameter affine model,

$$\begin{cases} C = F = \dfrac{v_{1x} - v_{0x}}{w} \\ E = -D = \dfrac{v_{1y} - v_{0y}}{w} \end{cases}$$

[0224] For 6-parameter affine model,

$$\begin{cases} C = \dfrac{v_{1x} - v_{0x}}{w} \\ D = \dfrac{v_{2x} - v_{0x}}{h} \\ E = \dfrac{v_{1y} - v_{0y}}{w} \\ F = \dfrac{v_{2y} - v_{0y}}{h} \end{cases}$$

where (v_{0x}, v_{0y}), (v_{1x}, v_{1y}), (v_{2x}, v_{2y}) may be the top-left, top-right, and bottom-left control point motion vectors. w and h may be the width and height of the CU.

[0225] The luma prediction refinement $\Delta I(i,j)$ may be added to the sub-block prediction $I(i,j)$. The prediction I' may be generated as the following equation:

$$I'(i,j) = I(i,j) + \Delta I(i,j).$$

[0226] PROF may not be applied in for an affine coded CU if: one or more (e.g., all) control point MVs are the same, which indicates the CU has translational motion; and/or the affine motion parameters are greater than a limit (e.g., a specified limit) because the sub-block based affine motion compensation (MC) is degraded to CU based MC to avoid large memory access bandwidth.

[0227] An encoding method may be applied, e.g., to reduce the encoding complexity of affine motion estimation with PROF. PROF may not be applied at affine motion estimation stage if: the CU is not the root block and the parent block of the CU does not select the affine mode as the best mode, PROF may not be applied since the possibility for current CU to select the affine mode as best mode is low; and/or if the magnitude of four affine parameters (C, D, E, F) are smaller than a predefined threshold and the current picture is not a low delay picture, PROF may not be applied because the improvement introduced by PROF is small. As described herein, the affine motion estimation with PROF may be accelerated.

[0228] A device, such as an encoder and/or a decoder, may be configured to increase the coding efficiency of the IBC prediction mode, e.g., for the compression of camera-captured video content. The camera-captured video content may be referred to interchangeably as natural video and/or natural video content.

[0229] As described herein, an affine model for block vector representation may be introduced to enrich a block vector model supported in IBC mode. For example, such enriched representation of block vector information may have a compression (e.g., a mean of better compressing) associated with one or more picture parts by catching a variety (e.g., large variety) of geometric transform relationships that may exist between a predicted block and a reference block of the predicted block, such as a zoom, a rotation, a symmetry, and/or the like.

[0230] To enrich a block vector model supported in IBC mode, two or more (e.g., two) means described herein may be configured and/or used. For example, two or more of the following may apply.

[0231] Two control point block vectors may be associated to an IBC coding unit and may represent the block vector information at one or more spatial locations, e.g., corresponding to the top-left and top-right corners of the considered CU (e.g., similarly to Affine inter mode).

[0232] In addition to and/or alternatively, three control point motion vectors may be used for an affine block vector field representation (e.g., a richer affine block vector field representation).

[0233] Similar to one or more procedures in inter prediction, an IBC Affine BVP mode may be introduced to code affine BV information. The IBC Affine BVP mode may include the signaling of a BV prediction index in an affine IBC AMVP candidate list, e.g., together with one or two motion vector differences.

[0234] Similar to one or more procedures in inter prediction, an IBC affine merge mode may be introduced to code affine BV information. The IBC affine merge mode may include the signaling of an affine BV merge index, e.g., identifying an affine BV model in an affine IBC merge candidate list.

[0235] In examples, an IBC affine merge mode may be supported.

[0236] In examples, an affine-IBC-MBVD block vector coding mode may be introduced. The affine-IBC-MBVD block vector coding mode may be included in the IBC affine merge mode described herein, e.g., where from BV offset may be signaled through an index and alternatively and/or additively applied to the control point block vector (CPBV) of the considered affine IBC merge CU.

[0237] In examples, an IBC affine mode may introduce a BV representation at a sub-block level, such as a 4x4 sub-block level. A BV may be assigned to a sub block (e.g., a 4x4 sub-block) of the affine IBC coding unit.

[0238] In examples, pixel-based affine prediction may be performed for an affine IBC CU. A BV may be assigned to a luma sample position inside the CU.

[0239] In examples, PROF may be applied to an affine IBC CU, e.g., after a sub-block-based (e.g., a 4x4 sub-block-based) affine prediction is performed.

[0240] The affine IBC mode described herein may be activated for the coding of camera-captured video content (e.g., not graphical video content).

[0241] In examples, the control point block vectors of a given CU may be constrained (e.g., normatively constrained) to ensure that one or more (e.g., all) block vectors in the affine BV field of one or more (e.g., all) CUs point to a prediction block spatially located inside the IBC search area, e.g., illustrated in FIG. 7.

[0242] The means may include, with respect to fractional block vector representation, an IBC block vector that is coded at accuracy-level corresponding to ¼-pel precision, 1-pel, and/or 4-pel precision. In the IBC affine AMVP mode described herein, two or more motion vector differences may be coded for a given CU, e.g., according to the affine-BV model used. The first MVD may be coded at the accuracy level associated to the current CU. If the first MVD may be coded at the accuracy level associated to the current CU, the other BVD may be coded differentially coded, e.g., under the form of a BV difference to the first BVD. For example, the other BVD may be coded with a doubled precision level compared to the first BVD, if the first BVD is coded at 1-pel or 4-pel accuracy level.

[0243] As described herein, Affine motion compensation may be supported in IBC mode. For example, Table 1 illustrates an example coding unit syntax table, e.g., supporting affine motion compensation in IBC mode as described herein.

[0244] One or more syntax elements may be configured (e.g., added) as illustrated in Table 1, e.g., an indication, such as an ibc_affine_flag may be signaled, a type of affine model may be signaled, one or two additional block vector differences may be signaled.

[0245] In examples, an indication, such as an ibc_affine_flag, may be added to indicate the use of affine BV model in IBC AMVP mode. The indication, such as the ibc_affine_flag, may be signaled if IBC affine is allowed at sequence parameter set (SPS) level and/or for a block size higher or equal to 16 in a width and a height.

[0246] If the indication, such as the ibc_affine_flag is true, a type of affine model may be signaled. For example, if two affine BV models are allowed (e.g., a 4-parameter affine model and a 6-parameter affine mode) as in inter mode, a type of affine model may be signaled.

[0247] If IBC affine is active for current IBC AMVP CU, one or two additional block vector differences may be signaled, e.g., according to the BV affine model used (e.g., 4-param or 6-param).

[0248] The syntax element mvp_l0_flag shown in Table 1 may be used for non-affine IBC AMVP mode and/or affine IBC AMVP mode. For example, the syntax element mvp_l0_flag may be configured to indicate the block vector predictor used to predictively code the BV information of the current CU in IBC AMVP mode (e.g., affine or not affine).

Table 1: an example enriched coding unit syntax table

coding_unit(x0, y0, cbWidth, cbHeight, cqtDepth, treeType, modeType) {	**Descriptor**
if(sh_slice_type = = I && (cbWidth > 64 || cbHeight > 64))	
modeType = MODE_TYPE_INTRA	
chType = treeType = = DUAL_TREE_CHROMA ? 1 : 0	
if(sh_slice_type != I || sps_ibc_enabled_flag) {	
if(treeType != DUAL_TREE_CHROMA && ((!(cbWidth = = 4 && cbHeight = = 4) && modeType != MODE_TYPE_INTRA) || (sps_ibc_enabled_flag && cbWidth <= 64 && cbHeight <= 64)))	
cu_skip_flag[x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] = = 0 && sh_slice_type != I && !(cbWidth = = 4 && cbHeight = = 4) && modeType = = MODE_TYPE_ALL)	
pred_mode_flag	ae(v)
if(((sh_slice_type = = I && cu_skip_flag[x0][y0] = =0) || (sh_slice_type != I && (CuPredMode[chType][x0][y0] != MODE_INTRA || (((cbWidth = = 4 && cbHeight = = 4) || modeType = = MODE_TYPE_INTRA) && cu_skip_flag[x0][y0] = = 0)))) && cbWidth <= 64 && cbHeight <= 64 && modeType != MODE_TYPE_INTER && sps_ibc_enabled_flag && treeType != DUAL_TREE_CHROMA)	
pred_mode_ibc_flag	ae(v)
}	
if(CuPredMode[chType][x0][y0] = = MODE_INTRA && sps_palette_enabled_flag && cbWidth <= 64 && cbHeight <= 64 && cu_skip_flag[x0][y0] = = 0 && modeType != MODE_TYPE_INTER && ((cbWidth * cbHeight) > (treeType != DUAL_TREE_CHROMA ? 16 : 16 * SubWidthC * SubHeightC)) && (modeType != MODE_TYPE_INTRA || treeType != DUAL_TREE_CHROMA))	
pred_mode_plt_flag	ae(v)

if(CuPredMode[chType][x0][y0] = = MODE_INTRA && sps_act_enabled_flag && treeType = = SINGLE_TREE)	
cu_act_enabled_flag	ae(v)
if(CuPredMode[chType][x0][y0] = = MODE_INTRA || CuPredMode[chType][x0][y0] = = MODE_PLT) {	
[...]	
} else if(treeType != DUAL_TREE_CHROMA) { /* MODE_INTER or MODE_IBC */	
if(cu_skip_flag[x0][y0] = = 0)	
general_merge_flag[x0][y0]	ae(v)
if(general_merge_flag[x0][y0])	
merge_data(x0, y0, cbWidth, cbHeight, chType)	
else if(CuPredMode[chType][x0][y0] = = MODE_IBC) {	
if(sps_ibc_affine_enabled_flag && cbWidth >= 16 && cbHeight >= 16) {	
ibc_affine_flag[x0][y0]	ae(v)
}	
if(sps_6param_ibc_affine_enabled_flag && ibc_affine_flag[x0][y0])	
cu_affine_type_flag[x0][y0]	ae(v)
mvd_coding(x0, y0, 0, 0)	
if(MotionModelIdc[x0][y0] > 0)	
mvd_coding(x0, y0, 0, 1)	
if(MotionModelIdc[x0][y0] > 1)	
mvd_coding(x0, y0, 0, 2)	
if(MaxNumIbcMergeCand > 1)	
mvp_l0_flag[x0][y0]	ae(v)
if(sps_amvr_enabled_flag && (MvdL0[x0][y0][0] != 0 || MvdL0[x0][y0][1] != 0))	
amvr_precision_idx[x0][y0]	ae(v)
} else {	
if(sh_slice_type = = B)	
inter_pred_idc[x0][y0]	ae(v)
if(sps_affine_enabled_flag && cbWidth >= 16 && cbHeight >= 16) {	
inter_affine_flag[x0][y0]	ae(v)
if(sps_6param_affine_enabled_flag && inter_affine_flag[x0][y0])	
cu_affine_type_flag[x0][y0]	ae(v)
}	
if(sps_smvd_enabled_flag && !ph_mvd_l1_zero_flag && inter_pred_idc[x0][y0] = = PRED_BI && !inter_affine_flag[x0][y0] && RefIdxSymL0 > −1 && RefIdxSymL1 > −1)	
sym_mvd_flag[x0][y0]	ae(v)
if(inter_pred_idc[x0][y0] != PRED_L1) {	
if(NumRefIdxActive[0] > 1 && !sym_mvd_flag[x0][y0])	

ref_idx_l0[x0][y0]	ae(v)
mvd_coding(x0, y0, 0, 0)	
if(MotionModelIdc[x0][y0] > 0)	
mvd_coding(x0, y0, 0, 1)	
if(MotionModelIdc[x0][y0] > 1)	
mvd_coding(x0, y0, 0, 2)	
mvp_l0_flag[x0][y0]	ae(v)
} else {	
MvdL0[x0][y0][0] = 0	
MvdL0[x0][y0][1] = 0	
}	
if(inter_pred_idc[x0][y0] != PRED_L0) {	
if(NumRefIdxActive[1] > 1 && !sym_mvd_flag[x0][y0])	
ref_idx_l1[x0][y0]	ae(v)
if(ph_mvd_l1_zero_flag && inter_pred_idc[x0][y0] = = PRED_BI) {	
MvdL1[x0][y0][0] = 0	
MvdL1[x0][y0][1] = 0	
MvdCpL1[x0][y0][0][0] = 0	
MvdCpL1[x0][y0][0][1] = 0	
MvdCpL1[x0][y0][1][0] = 0	
MvdCpL1[x0][y0][1][1] = 0	
MvdCpL1[x0][y0][2][0] = 0	
MvdCpL1[x0][y0][2][1] = 0	
} else {	
if(sym_mvd_flag[x0][y0]) {	
MvdL1[x0][y0][0] = −MvdL0[x0][y0][0]	
MvdL1[x0][y0][1] = −MvdL0[x0][y0][1]	
} else	
mvd_coding(x0, y0, 1, 0)	
if(MotionModelIdc[x0][y0] > 0)	
mvd_coding(x0, y0, 1, 1)	
if(MotionModelIdc[x0][y0] > 1)	
mvd_coding(x0, y0, 1, 2)	
}	
mvp_l1_flag[x0][y0]	ae(v)
} else {	
MvdL1[x0][y0][0] = 0	
MvdL1[x0][y0][1] = 0	
}	

if((sps_amvr_enabled_flag && inter_affine_flag[x0][y0] = = 0 && (MvdL0[x0][y0][0] != 0 || MvdL0[x0][y0][1] != 0 || MvdL1[x0][y0][0] != 0 || MvdL1[x0][y0][1] != 0)) || (sps_affine_amvr_enabled_flag && inter_affine_flag[x0][y0] = = 1 && (MvdCpL0[x0][y0][0][0] != 0 || MvdCpL0[x0][y0][0][1] != 0 || MvdCpL1[x0][y0][0][0] != 0 || MvdCpL1[x0][y0][0][1] != 0 || MvdCpL0[x0][y0][1][0] != 0 || MvdCpL0[x0][y0][1][1] != 0 || MvdCpL1[x0][y0][1][0] != 0 || MvdCpL1[x0][y0][1][1] != 0 || MvdCpL0[x0][y0][2][0] != 0 || MvdCpL0[x0][y0][2][1] != 0 || MvdCpL1[x0][y0][2][0] != 0 || MvdCpL1[x0][y0][2][1] != 0))) {	
amvr_flag[x0][y0]	ae(v)
if(amvr_flag[x0][y0])	
amvr_precision_idx[x0][y0]	ae(v)
}	
if(sps_bcw_enabled_flag && inter_pred_idc[x0][y0] = = PRED_BI && luma_weight_l0_flag[ref_idx_l0 [x0][y0]] = = 0 && luma_weight_l1_flag[ref_idx_l1 [x0][y0]] = = 0 && chroma_weight_l0_flag[ref_idx_l0 [x0][y0]] = = 0 && chroma_weight_l1_flag[ref_idx_l1 [x0][y0]] = = 0 && cbWidth * cbHeight >= 256)	
bcw_idx[x0][y0]	ae(v)
}	
}	
[…]	
}	

[0249] Table 2 illustrates an example enriched BV data coding syntax to support affine IBC merge mode as described herein. As shown in Table 2, an indication, such as an ibc_merge_affine_flag, may signal the use of an affine BV representation for the current CU in IBC merge mode.

[0250] For example, the indication, such as the ibc_merge_affine_flag, may be followed by the IBC merge_idx syntax element. The ibc_merge_affine_flag may indicate to a device, such as a decoder, from which BV prediction candidate to derive BV data of the current CU. For example, the BV data may be derived in the IBC merge mode or in the IBC affine merge mode.

Table 2: an example enriched merge_data syntax table

merge_data(x0, y0, cbWidth, cbHeight, chType) {	**Descriptor**
if(CuPredMode[chType][x0][y0] = = MODE_IBC) {	
ibc_merge_affine_flag[x0][y0]	ae(v)
if(MaxNumIbcMergeCand > 1)	
merge_idx[x0][y0]	ae(v)
} else {	

if(MaxNumSub-blockMergeCand > 0 && cbWidth >= 8 && cbHeight >= 8)	
merge_sub-block_flag[x0][y0]	ae(v)
if(merge_sub-block_flag[x0][y0] = = 1) {	
if(MaxNumSub-blockMergeCand > 1)	
merge_sub-block_idx[x0][y0]	ae(v)
} else {	
if(cbWidth < 128 && cbHeight < 128 && ((sps_ciip_enabled_flag && cu_skip_flag[x0][y0] = = 0 && (cbWidth * cbHeight) >= 64) || (sps_gpm_enabled_flag && sh_slice_type = = B && cbWidth >= 8 && cbHeight >= 8 && cbWidth < (8 * cbHeight) && cbHeight < (8 * cbWidth))))	
regular_merge_flag[x0][y0]	ae(v)
if(regular_merge_flag[x0][y0] = = 1) {	
if(sps_mmvd_enabled_flag)	
mmvd_merge_flag[x0][y0]	ae(v)
if(mmvd_merge_flag[x0][y0] = = 1) {	
if(MaxNumMergeCand > 1)	
mmvd_cand_flag[x0][y0]	ae(v)
mmvd_distance_idx[x0][y0]	ae(v)
mmvd_direction_idx[x0][y0]	ae(v)
} else if(MaxNumMergeCand > 1)	
merge_idx[x0][y0]	ae(v)
} else {	
if(sps_ciip_enabled_flag && sps_gpm_enabled_flag && sh_slice_type = = B && cu_skip_flag[x0][y0] = = 0 && cbWidth >= 8 && cbHeight >= 8 && cbWidth < (8 * cbHeight) && cbHeight < (8 * cbWidth) && cbWidth < 128 && cbHeight < 128)	
ciip_flag[x0][y0]	ae(v)
if(ciip_flag[x0][y0] && MaxNumMergeCand > 1)	
merge_idx[x0][y0]	ae(v)
if(!ciip_flag[x0][y0]) {	
merge_gpm_partition_idx[x0][y0]	ae(v)
merge_gpm_idx0[x0][y0]	ae(v)
if(MaxNumGpmMergeCand > 2)	
merge_gpm_idx1[x0][y0]	ae(v)
}	
}	
}	
}	

}	

[0251] FIG. 15 illustrates an example IBC affine CU decoding procedure as described herein. As illustrated in FIG. 15, a CU may be checked to determine if the CU is in IBC mode and/or Affine mode. If the CU is not in the IBC mode and/or the Affine mode, a decoding procedure may be performed.

[0252] As illustrated in FIG. 15, if the CU is in IBC mode, e.g., an IBC CU, whether a CU is in merge mode may be determined.

[0253] If the CU is not in merge mode, the CU may be in IBC affine AMVP mode. The list of affine IBC control point block vector predictor candidates for current CU may be constructed. The list of affine IBC control point block vector predictor candidates may include an affine AMVP candidate list construction (e.g., similar to the inter prediction).

[0254] Based on the decoder mvp_l0 syntax element, e.g., as shown in Table 1, the selected predictor of CPVC of current CU may be obtained.

[0255] The decoded BV difference associated with a CPBV may be added to the corresponding BVs. The addition may produce the decoded CPBVs of current CU.

[0256] If an affine IBC CU is in merge mode, the list of affine IBC merge candidates may be constructed (e.g., similar to the affine merge candidate list construction of inter prediction but applied to block vectors).

[0257] The CPBVs of current CU may be derived as the CPBVs indicated by the parsed merge_idx syntax element.

[0258] For AMVP coding mode and/or merge affine IBC coding mode, if the CPBVs of current CU are obtained, the BV affine field of current CU may be computed (e.g., the same way as the affine motion field computation in inter affine case).

[0259] Affine motion compensation may be applied, e.g., producing the prediction block of current CU.

[0260] The prediction refinement based on optical flow (PROF) may be applied to the predicted CU, e.g., to enhance the quality of the predicted block. The applied PROF may be similar (e.g., the same) process as the one for inter prediction.

[0261] The residual block of the current CU and/or the predicted block may be added to obtained decoded IBC affine CU.

[0262] The two control point block vectors mode may be supported for IBC, e.g., corresponding to the 4-parameter affine model of inter prediction.

[0263] In examples, IBC affine merge mode may be supported (e.g., only IBC affine merge mode may be supported). The support of IBC affine merge mode may achieve compression efficiency and/or reduce complexity (e.g., compared to the case where affine IBC AMVP mode is supported).

[0264] In examples, alternatively and/or in addition, Affine AMVP mode may be used in one or more pictures. For example, Affine AMVP mode may be used in an inter picture with low temporal layer ID (e.g., the lowest temporal layer ID). The inter picture with low temporal layer ID (e.g., the lowest temporal layer ID) may be used as a reference picture by one or more other pictures.

[0265] In examples, an affine-IBC-MBVD block vector coding mode may be employed. An affine-IBC-MBVD block vector coding mode may be included in the IBC affine merge mode described herein. For example, a BV offset may be signaled through an index and additively and/or alternatively applied to the CPBV of the considered affine IBC merge CU.

[0266] In examples, pixel-based affine prediction may be performed for an affine IBC CU. A BV may be assigned to a luma sample position inside the CU. Compression efficiency may be achieved.

[0267] If pixel-based affine model is used for IBC, PROF prediction refinement may be skipped during the prediction of a CU.

[0268] The affine IBC mode described herein may be activated for the coding of camera-captured video content (e.g., and not graphical video content).

[0269] In examples, the control point block vectors of a given CU may be constrained (e.g., normatively constrained) that one or more (e.g., all) block vectors in the affine BV field of one or more (e.g., all) CUs point to a prediction block spatially located inside the IBC search area, e.g., illustrated in FIG. 7.

[0270] With respect to fractional block vector representation, an IBC block vector may be coded at accuracy-level corresponding to ¼-pel precision, 1-pel, and/or 4-pel precision. In the IBC affine AMVP mode described herein, two or more motion vector differences may be coded for a given CU, e.g., according to the affine-BV model used. The first MVd may be coded at the accuracy level associated to the current CU. If the first MVd is coded at the accuracy level associated to the current CU, other BVd of the affine CU may be coded differentially coded in comparison to the first BVd. For example, the other BVd may be coded with a doubled precision level compared to the first BVd, if the first BVd is coded at 1-pel or 4-pel accuracy level.

[0271] In examples, affine block vector computed on a sub-block basis may be stored in the motion data storing buffer of the video codec used, e.g., typically on a 4x4 block basis. The computed affine block vector stored in the buffer may be used for the spatial prediction of the block vector data of future block in the same picture, which may be in affine mode or non-affine mode.

[0272] In examples, affine block vector computed on a sub-block basis may be stored in the motion data storing buffer of the video codec used, e.g., typically on a 4x4 block basis. The computed affine block vector stored in the buffer may be used for the temporal prediction of the block vector data of block in one or more future pictures, e.g., which may use the current picture as a reference picture for temporal prediction.

[0273] In examples, the IBC affine motion described herein may be controlled at high level, e.g., at a sequence level. For example, the IBC affine motion described herein may be controlled using a dedicated SPS signaled indication, such as a dedicated SPS signaled flag.

[0274] In examples, the IBC affine motion described herein may be controlled at a high level, e.g., at a picture level. For example, the IBC affine motion described herein may be controlled using a dedicated picture header signaled indication, such as a dedicated picture header signaled flag.

[0275] In examples, the IBC affine motion described herein may be controlled at a high level, e.g., at a slice level. For example, the IBC affine motion described herein may be controlled using a dedicated slice header signaled indication, such as a dedicated slice header signaled flag.

[0276] In examples, the IBC affine motion described herein may be controlled at a high level, e.g., at a sub-picture level, a tile level, and/or a tile group level.

[0277] Although features and elements are described above in particular combinations, one of ordinary skill in the art will appreciate that each feature or element can be used alone or in any combination with the other features and elements. In addition, the methods described herein may be implemented in a computer program, software, or firmware incorporated in a computer-readable medium for execution by a computer or processor. Examples of computer-readable media include electronic signals (transmitted over wired or wireless connections) and computer-readable storage media. Examples of computer-readable storage media include, but are not limited to, a read only memory (ROM), a random access memory (RAM), a register, cache memory, semiconductor memory devices, magnetic media such as internal hard disks and removable disks, magneto-optical media, and optical media such as CD-ROM disks, and digital versatile disks (DVDs). A processor in association with software may be used to implement a radio frequency transceiver for use in a WTRU, UE, terminal, base station, RNC, or any host computer.

CLAIMS

1. A device for video decoding, the device comprising:

a processor configured to:

obtain an affine intra block copy (IBC) mode indication in video data;

based on the obtained affine IBC mode indication, determine that a current block is associated with an affine IBC;

based on the determination that the current block is associated with the affine IBC, obtain a plurality of control point block vectors (CPBVs) associated with the current block; and

decode the current block based on the plurality of CPBVs.

2. A method for video decoding, the method comprising:

obtaining an affine intra block copy (IBC) mode indication in video data;

based on the obtained affine IBC mode indication, determining that a current block is associated with an affine IBC;

based on the determination that the current block is associated with the affine IBC, obtaining a plurality of control point block vectors (CPBVs) associated with the current block; and

decoding the current block based on the plurality of CPBVs.

3. A device for video encoding, the device comprising:

a processor configured to:

obtain a current block associated with a video content;

determine whether the current block is associated with an affine intra block copy (IBC) mode;

based on a determination that the current block is associated with the IBC mode, obtain plurality of control point block vectors (CPBVs) associated with the current block;

encode the current block based on the plurality of CPBVs.

4. A method for video encoding, the method comprising:

obtaining a current block associated with a video content;

determining whether the current block is associated with an affine intra block copy (IBC) mode;

based on a determination that the current block is associated with the IBC mode, obtaining plurality of control point block vectors (CPBVs) associated with the current block;

encoding the current block based on the plurality of CPBVs.

5. The device of claim 1 or claim 3, wherein the processor is configured to:

determine that the current block is associated with a merge IBC mode; and

based on the determination that the current block is associated with the merge IBC mode, construct an IBC affine control point block vector (BV) prediction candidate list, wherein the plurality of CPBVs is obtained based on the BV prediction candidate list.

6. The device of claim 1 or claim 3, wherein the processor is configured to:

apply affine motion compensation based on the plurality of CPBVs to obtain a prediction block; and

apply prediction refinement based on optical flow (PROF) to the obtained prediction block.

7. The method of claim 2 or claim 4, wherein the method comprises:

determining that the current block is associated with a merge IBC mode; and

based on the determination that the current block is associated with the merge IBC mode, constructing an IBC affine control point block vector (BV) prediction candidate list, wherein the plurality of CPBVs is obtained based on the BV prediction candidate list.

8. The method of claim 2 or claim 4, wherein the method comprises:

applying affine motion compensation based on the plurality of CPBVs to obtain a prediction block; and

applying prediction refinement based on optical flow (PROF) to the obtained prediction block.

9. The method of claim 4, wherein the method comprises:

based on a determination that the current block is associated with the IBC mode, including an affine IBC mode indication in video data.

10. The device of claim 3, wherein the processor is configured to:

based on a determination that the current block is associated with the IBC mode, include an affine IBC mode indication in video data.

11. A computer readable storage medium including instructions for video decoding, causing a processor to perform the method of any one of claims 2, 7, or 8.

12. A computer readable storage medium including instructions for video decoding, causing a processor to perform the method of any one of claims 4, 7, 8, or 9.

13. A non-transitory computer readable storage medium including instructions for video decoding, causing a processor to perform the method of any one of claims 2, 7, or 8.

14. A non-transitory computer readable storage medium including instructions for video decoding, causing a processor to perform the method of any one of claims 4, 7, 8, or 9.

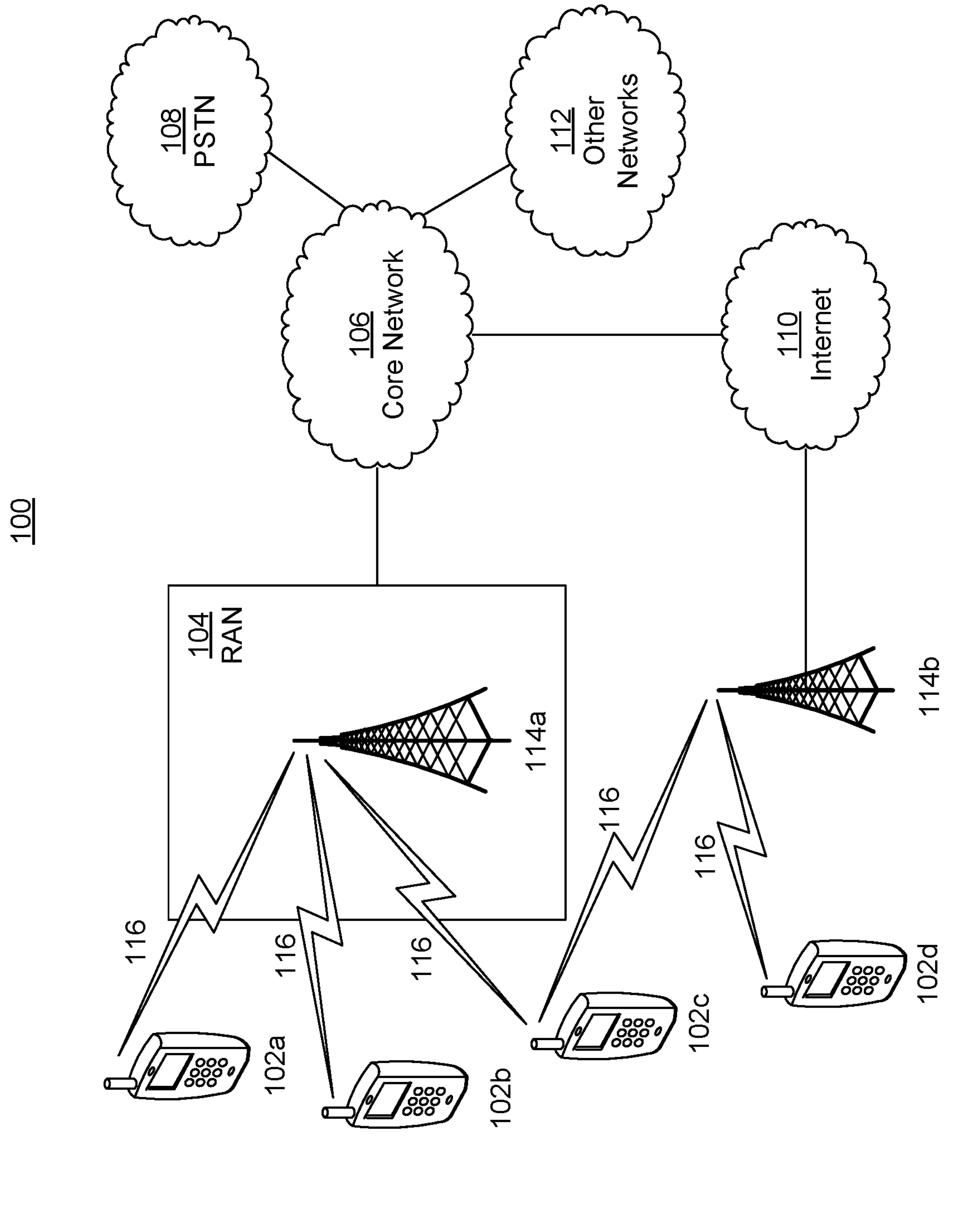
100
108
PSTN
112
Other
Networks
106
Core Network
110
Internet
104
RAN
114a
114b
116
116
116
116
116
102a
102b
102c
102d

FIG. 1A

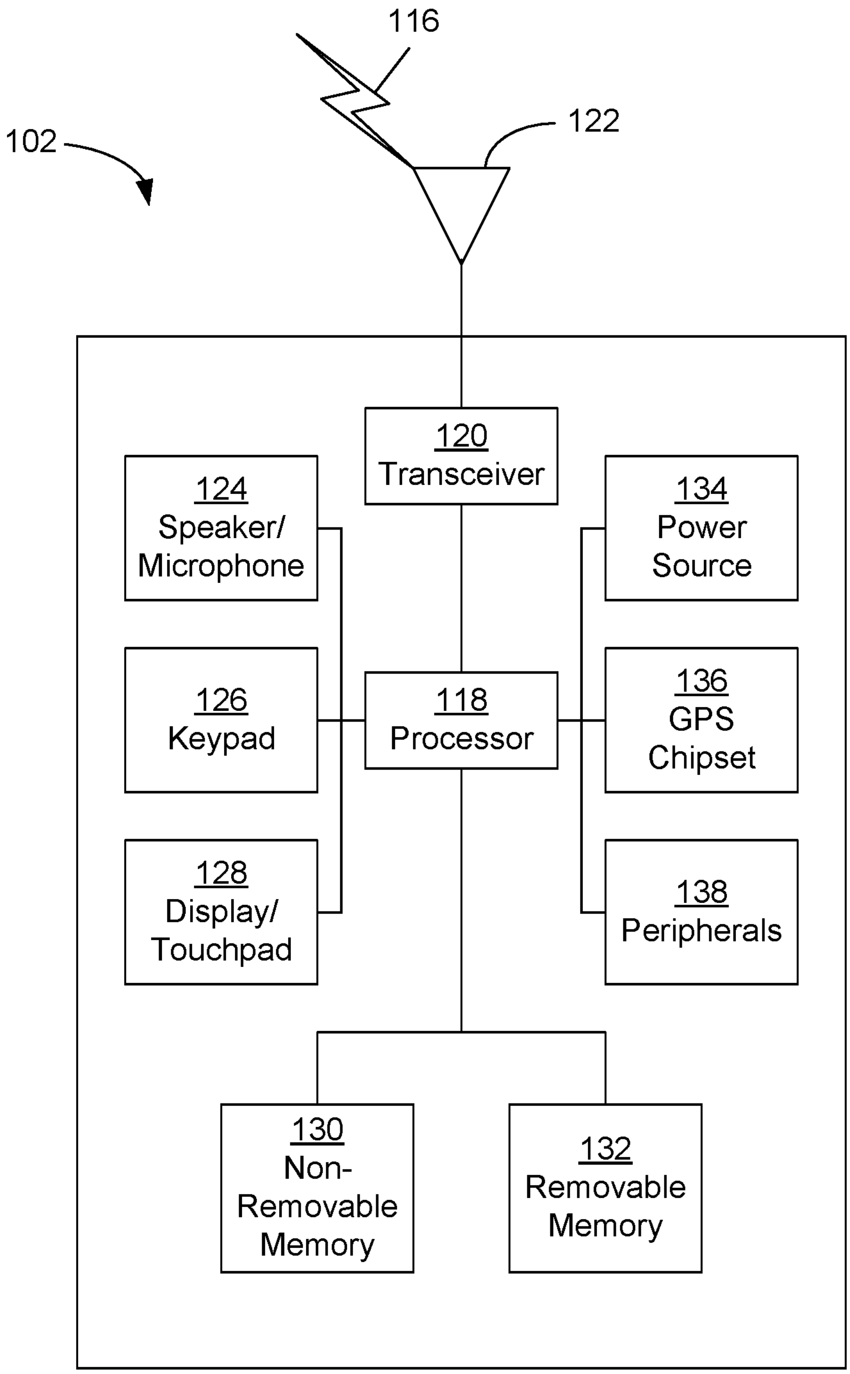
116
122
102
120
Transceiver
124
Speaker/
Microphone
134
Power
Source
126
Keypad
118
Processor
136
GPS
Chipset
128
Display/
Touchpad
138
Peripherals
130
Non-
Removable
Memory
132
Removable
Memory

FIG. 1B

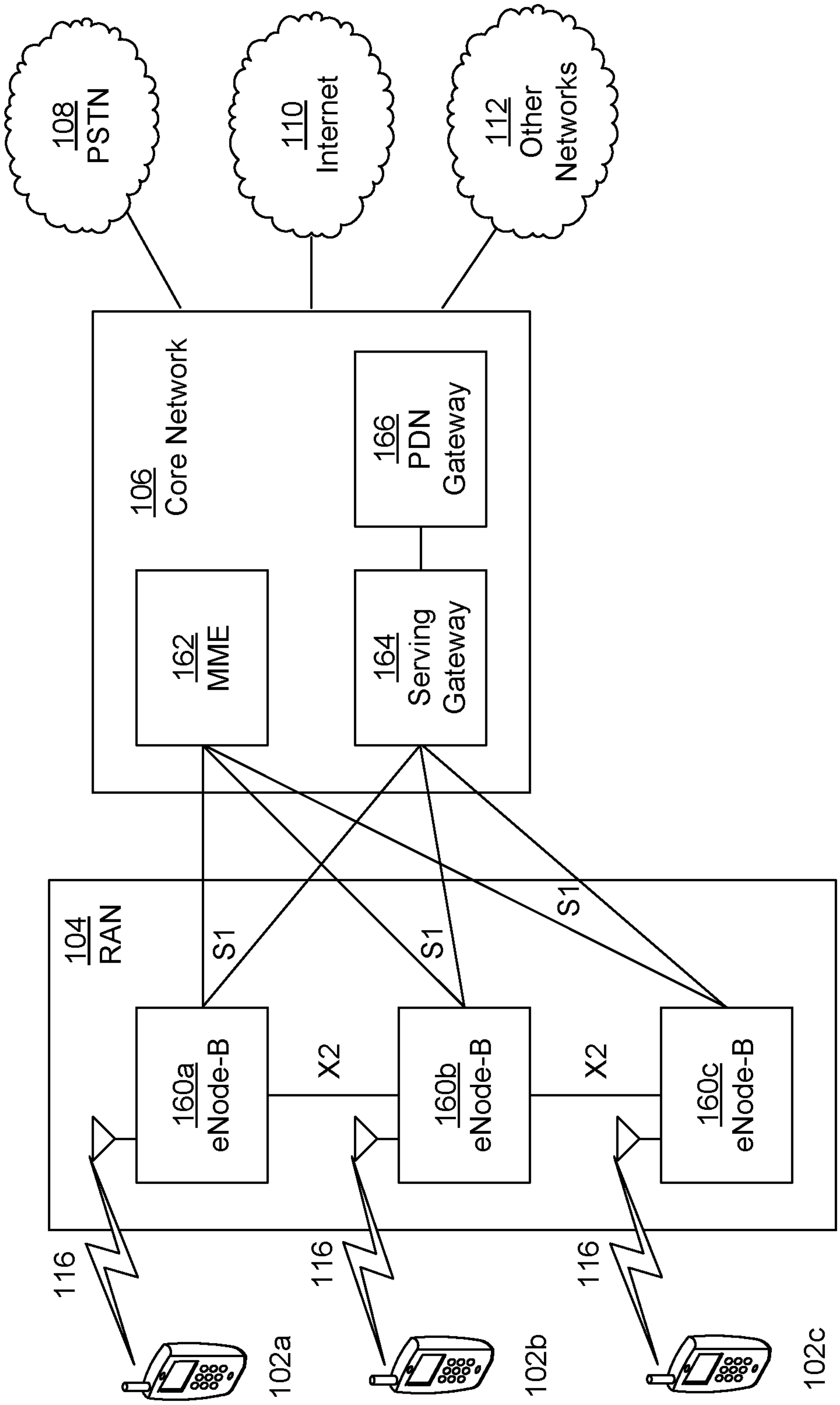
108
PSTN
110
Internet
112
Other
Networks
106
Core Network
162
MME
164
Serving
Gateway
166
PDN
Gateway
104
RAN
S1
S1
S1
160a
eNode-B
160b
eNode-B
160c
eNode-B
X2
X2
116
116
116
102a
102b
102c

FIG. 1C

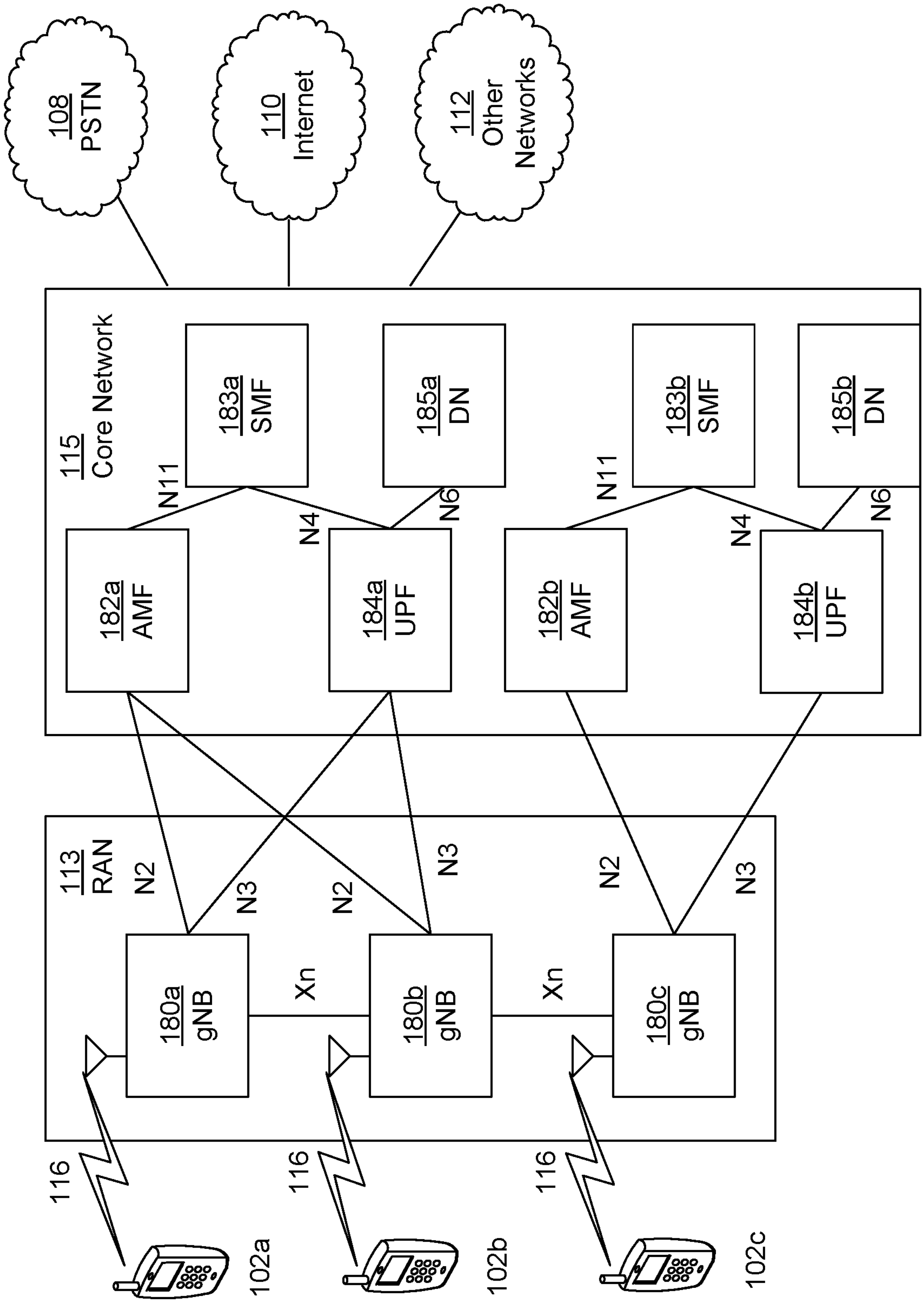
108
PSTN
110
Internet
112
Other
Networks
115
Core Network
183a
SMF
185a
DN
183b
SMF
185b
DN
182a
AMF
184a
UPF
182b
AMF
184b
UPF
N11
N4
N6
N11
N4
N6
113
RAN
N2
N3
N2
N3
N2
N3
180a
gNB
180b
gNB
180c
gNB
Xn
Xn
116
116
116
102a
102b
102c

FIG. 1D

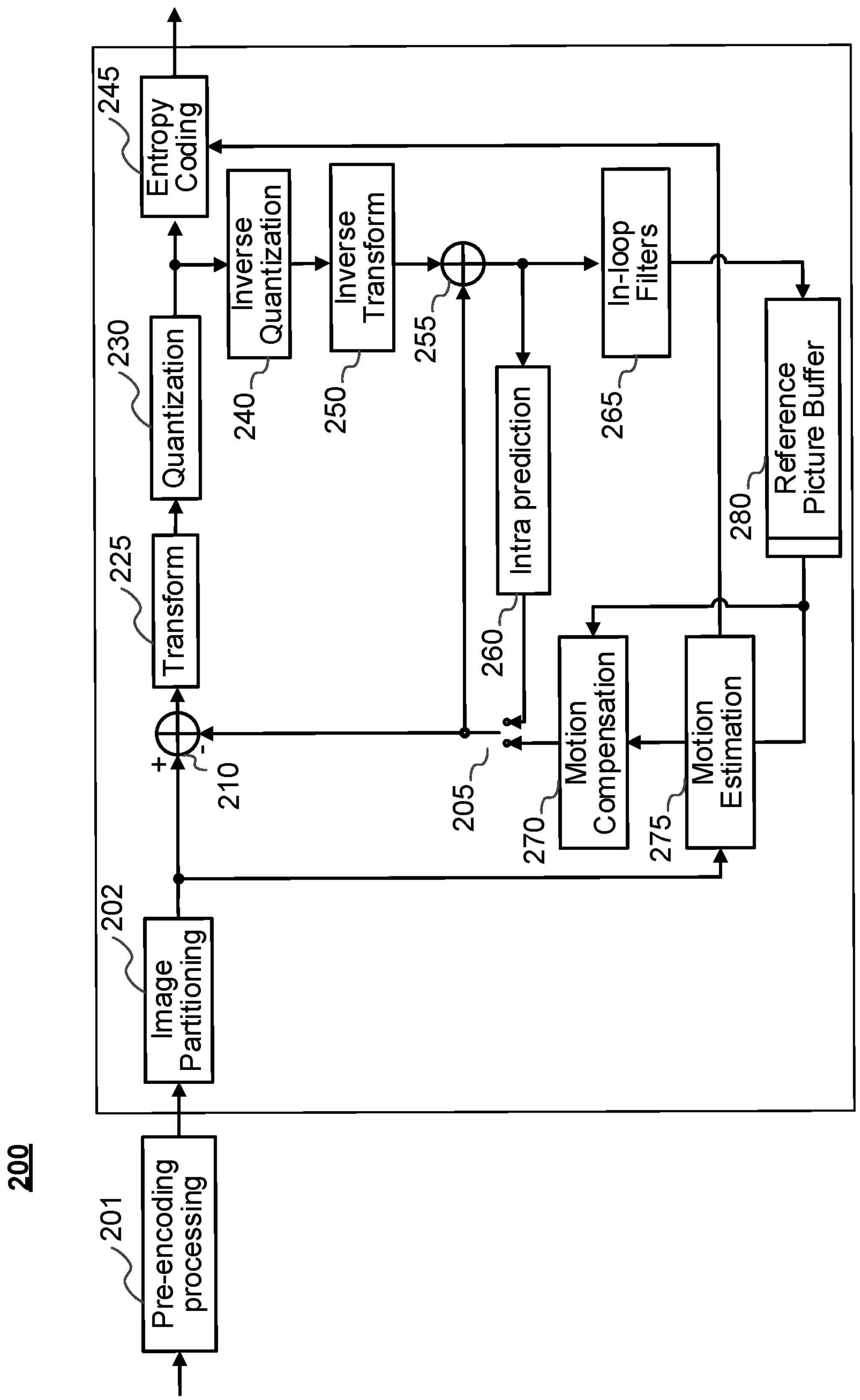
200
201
Pre-encoding processing
202
Image Partitioning
210
+
-
225
Transform
230
Quantization
245
Entropy Coding
240
Inverse Quantization
250
Inverse Transform
255
260
Intra prediction
265
In-loop Filters
205
270
Motion Compensation
275
Motion Estimation
280
Reference Picture Buffer

FIG. 2

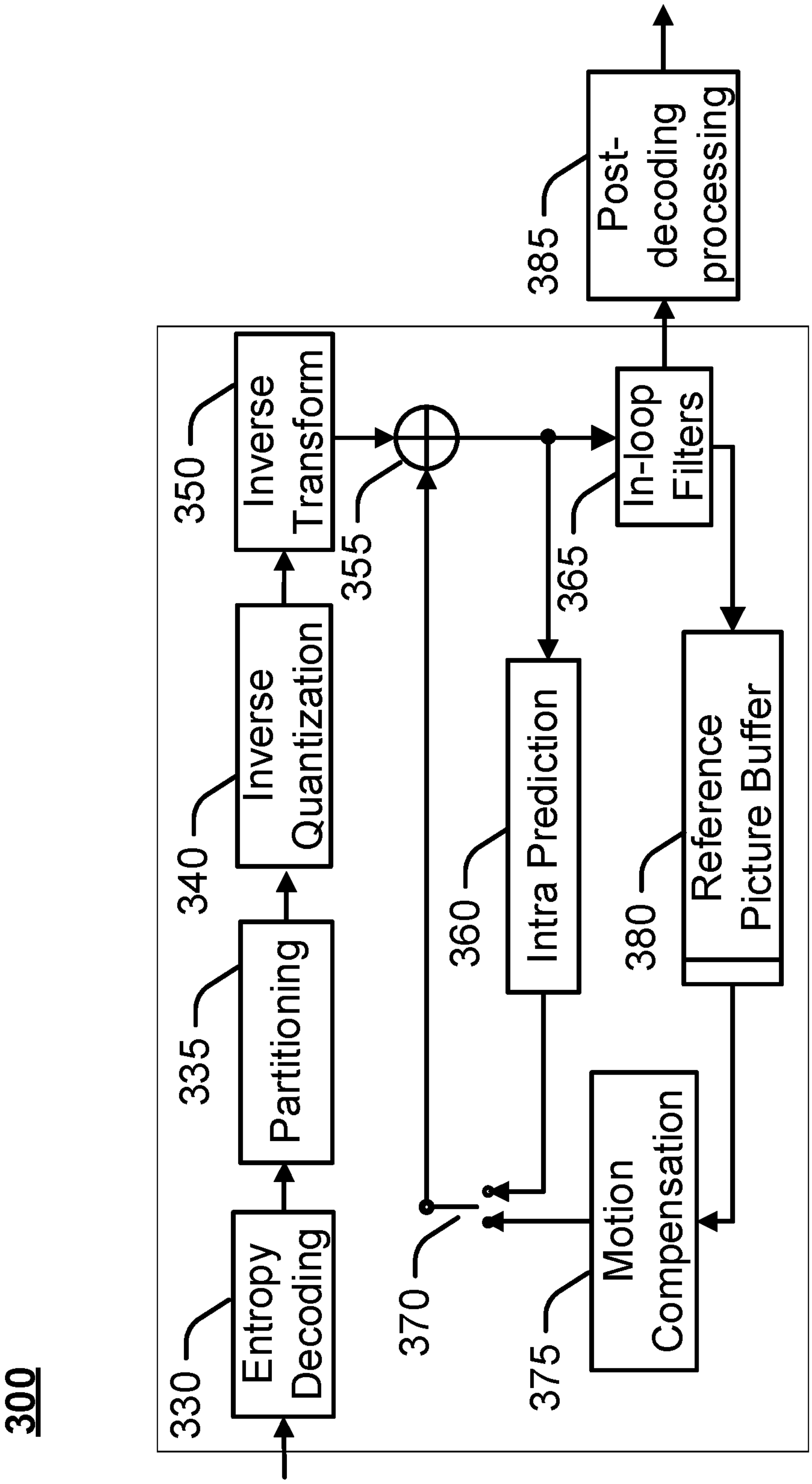
300
330
Entropy Decoding
335
Partitioning
340
Inverse Quantization
350
Inverse Transform
355
365
In-loop Filters
385
Post-decoding processing
360
Intra Prediction
380
Reference Picture Buffer
370
375
Motion Compensation

FIG. 3

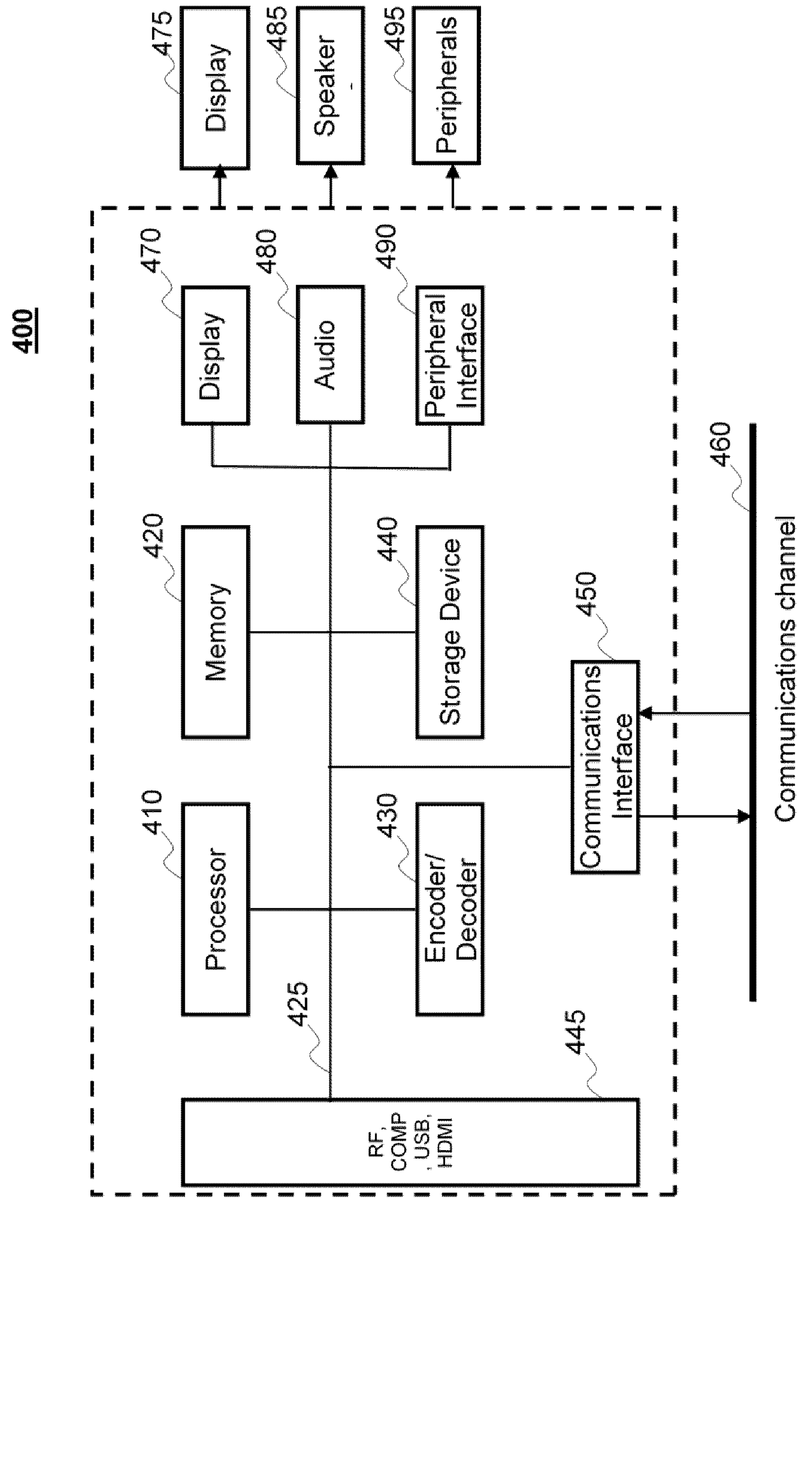
400
Display
475
Speaker
485
Peripherals
495
Display
470
Audio
480
Peripheral Interface
490
Memory
420
Storage Device
440
Processor
410
Encoder/ Decoder
430
Communications Interface
450
Communications channel
460
425
RF, COMP, USB, HDMI
445

FIG. 4

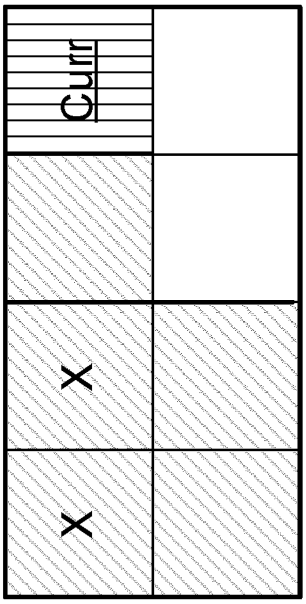

FIG. 5B

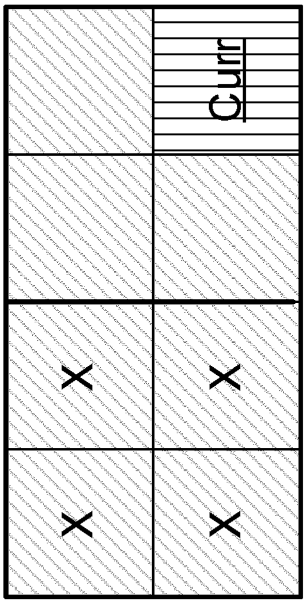

FIG. 5D

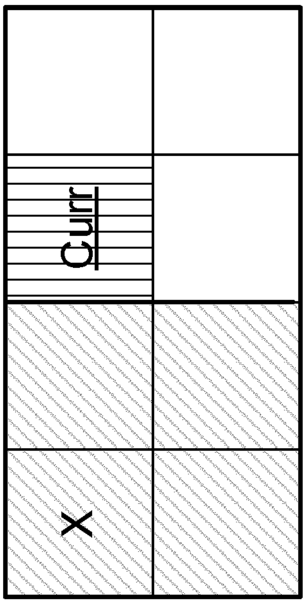

FIG. 5A

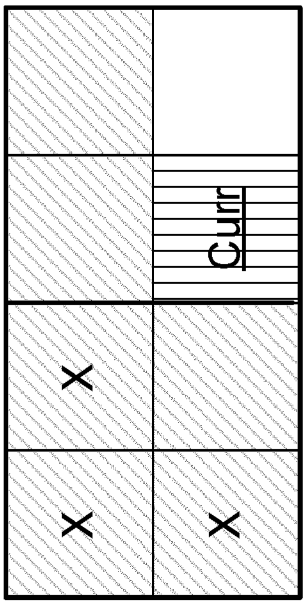

FIG. 5C

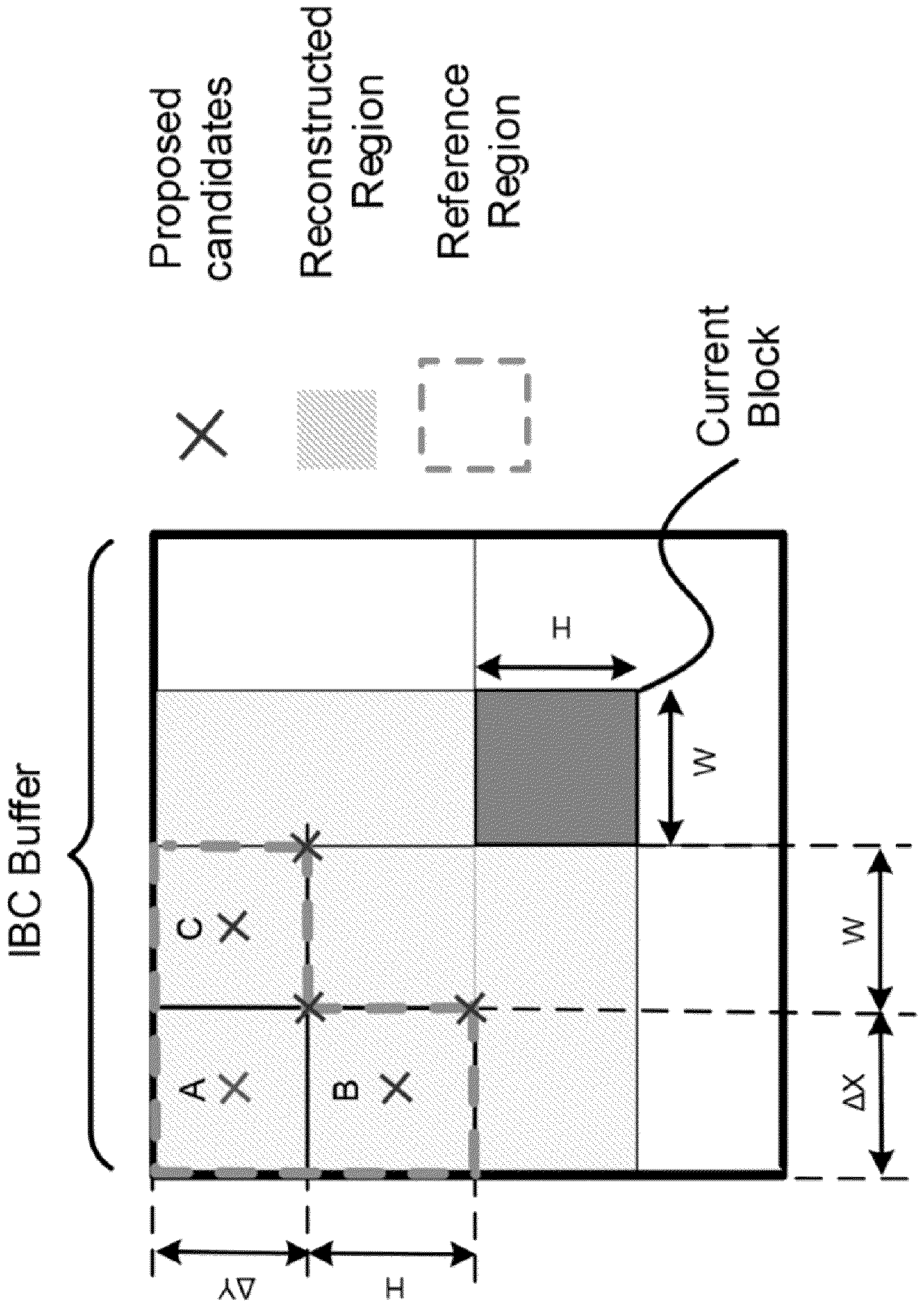
IBC Buffer
Proposed candidates
Reconstructed Region
Reference Region
Current Block
H
W
W
ΔX
ΔY
H
A
B
C

FIG. 6

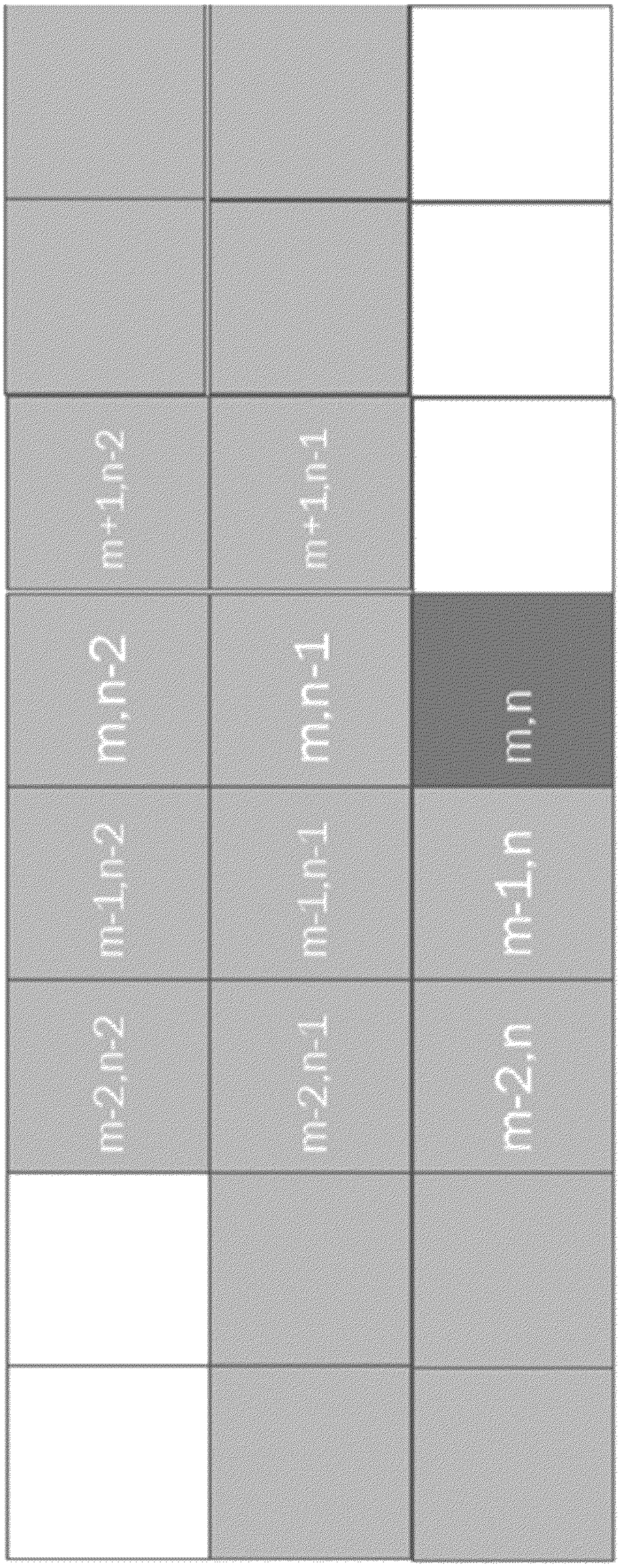

FIG. 7

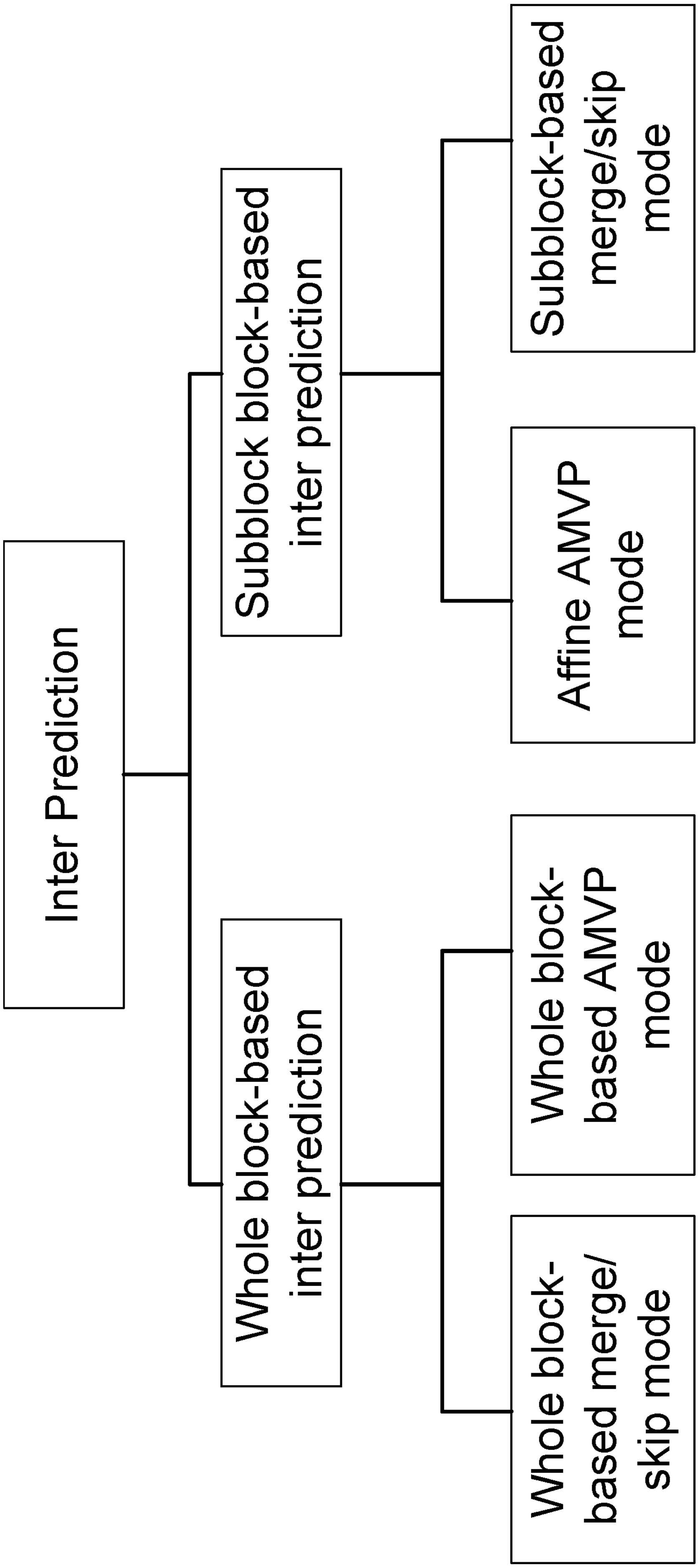
Inter Prediction
Whole block-based inter prediction
Subblock block-based inter prediction
Whole block-based merge/skip mode
Whole block-based AMVP mode
Affine AMVP mode
Subblock-based merge/skip mode

FIG. 8

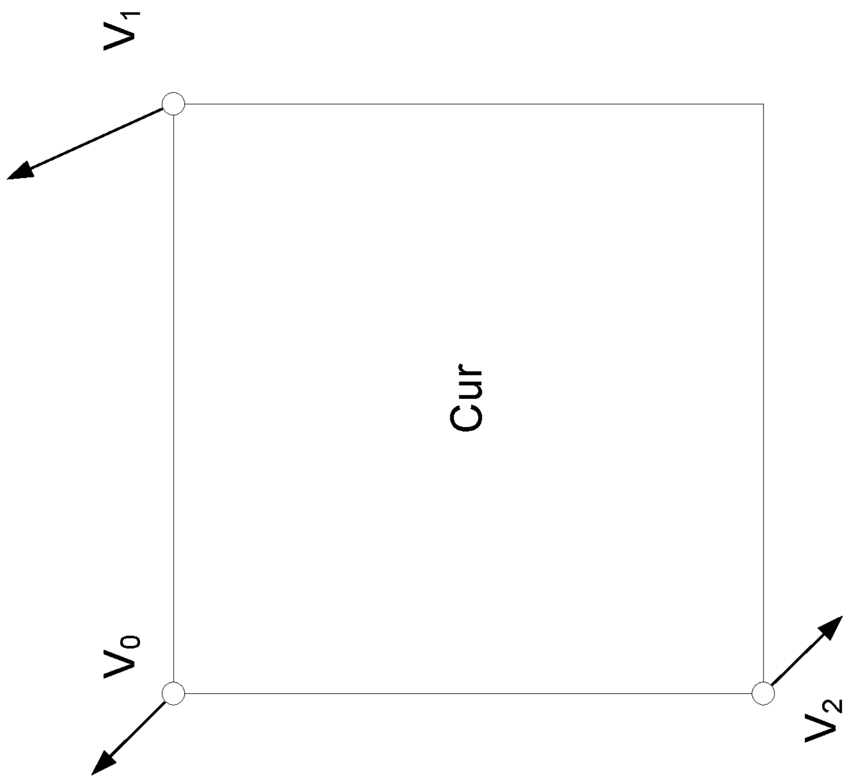

FIG. 9B

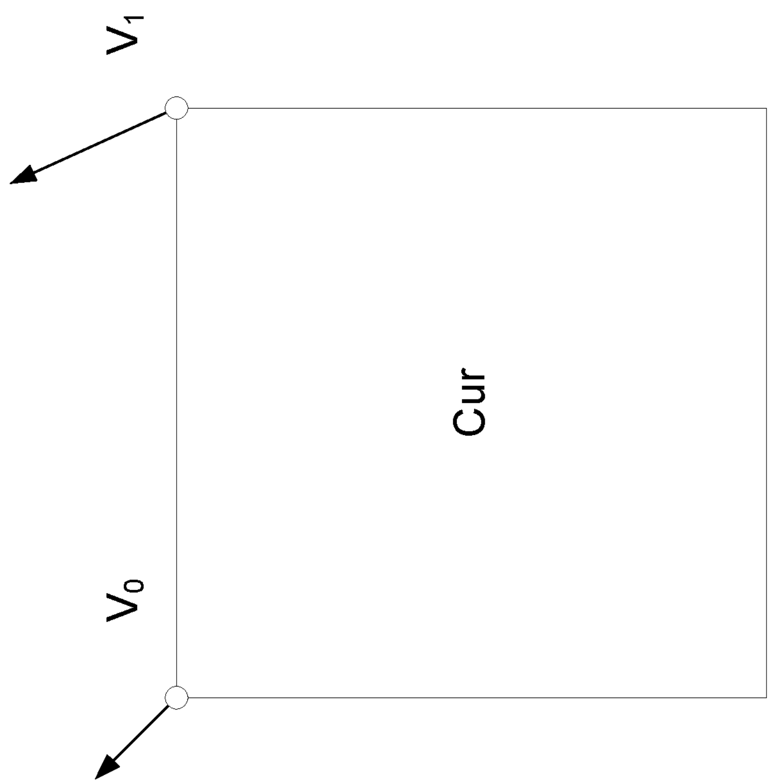

FIG. 9A

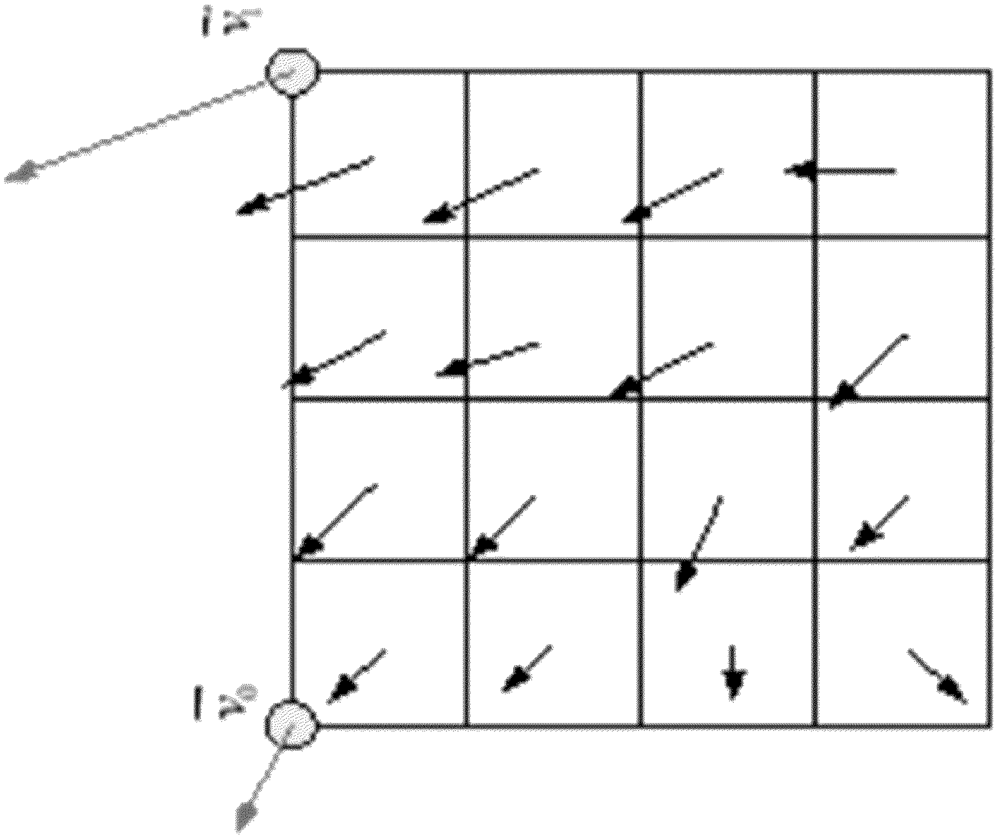

FIG. 10

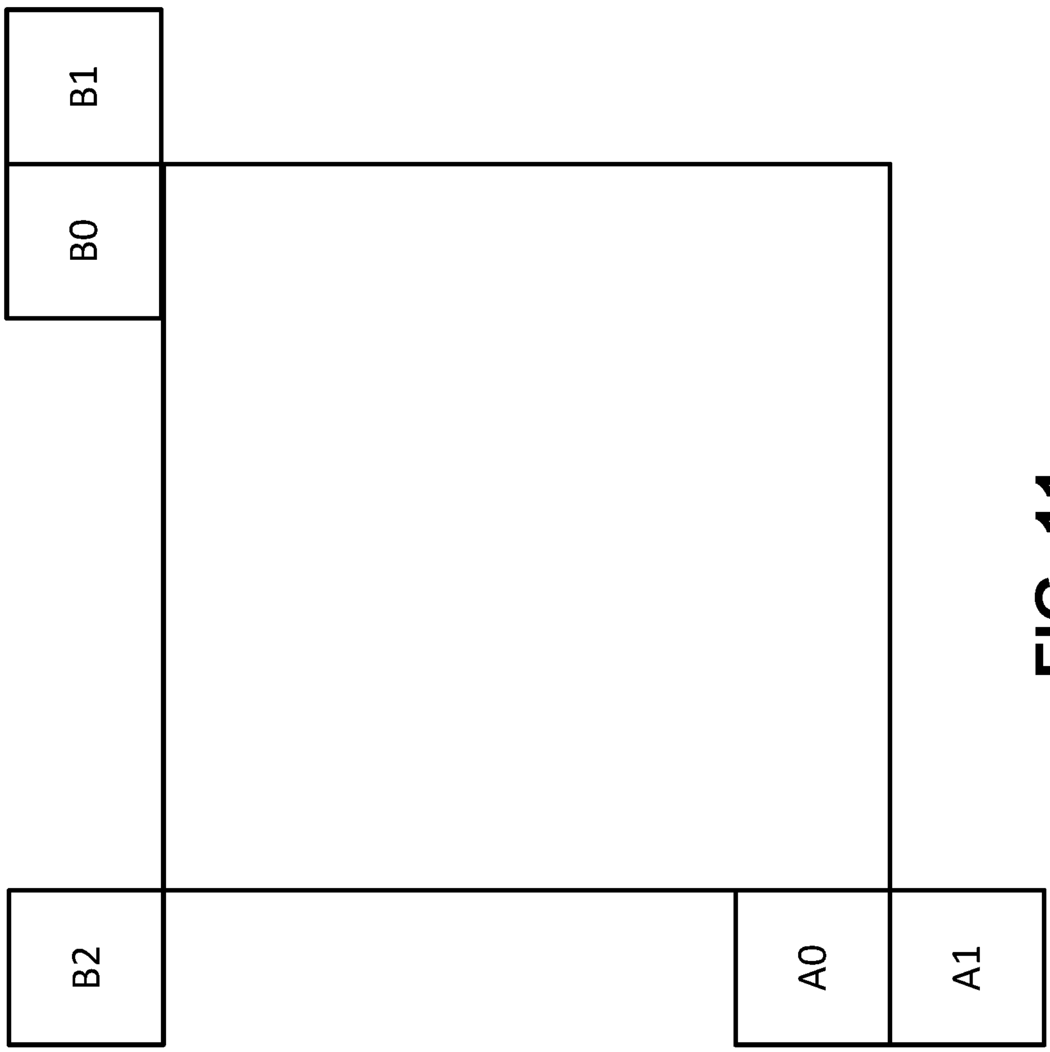

FIG. 11

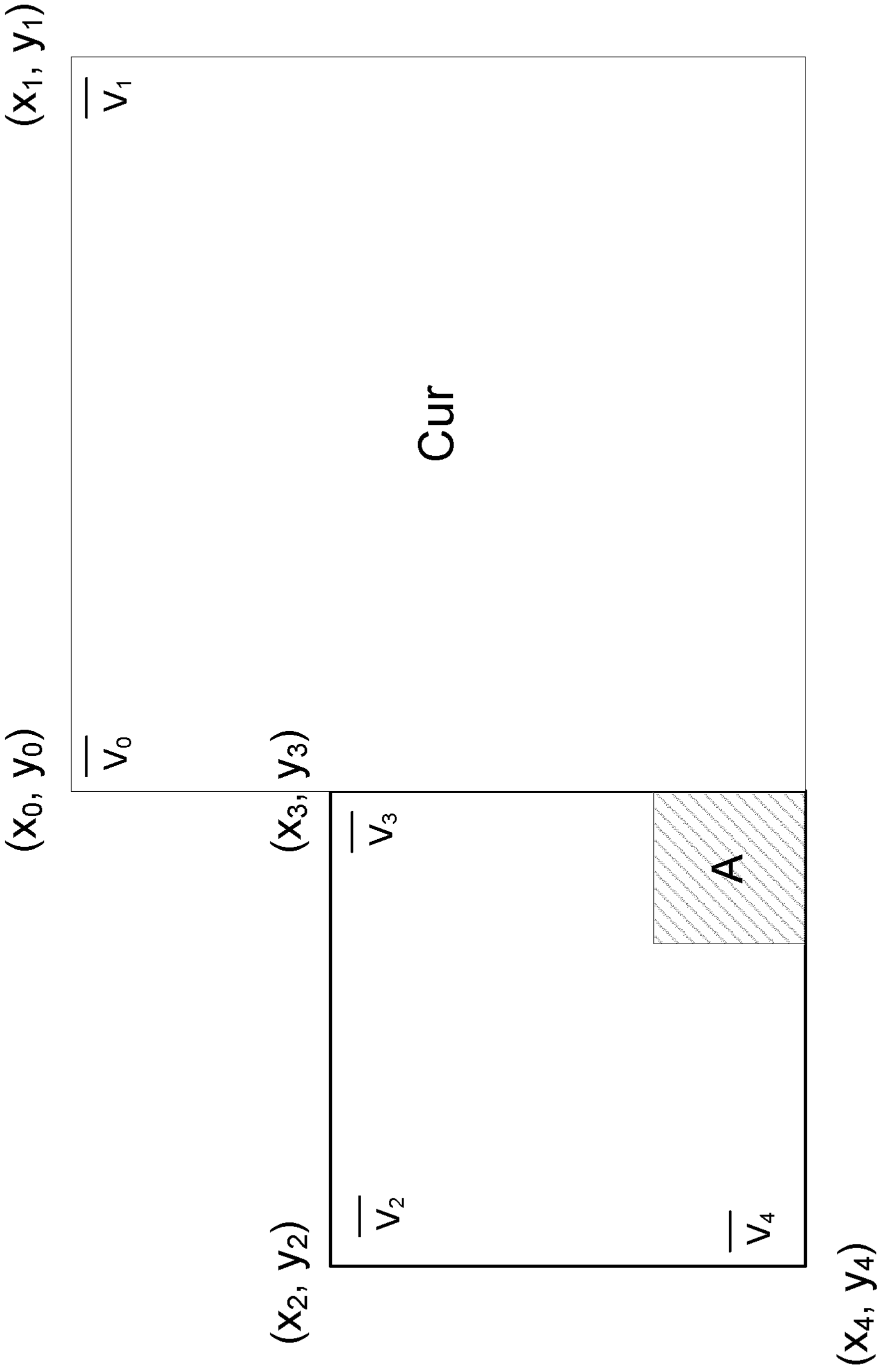

FIG. 12

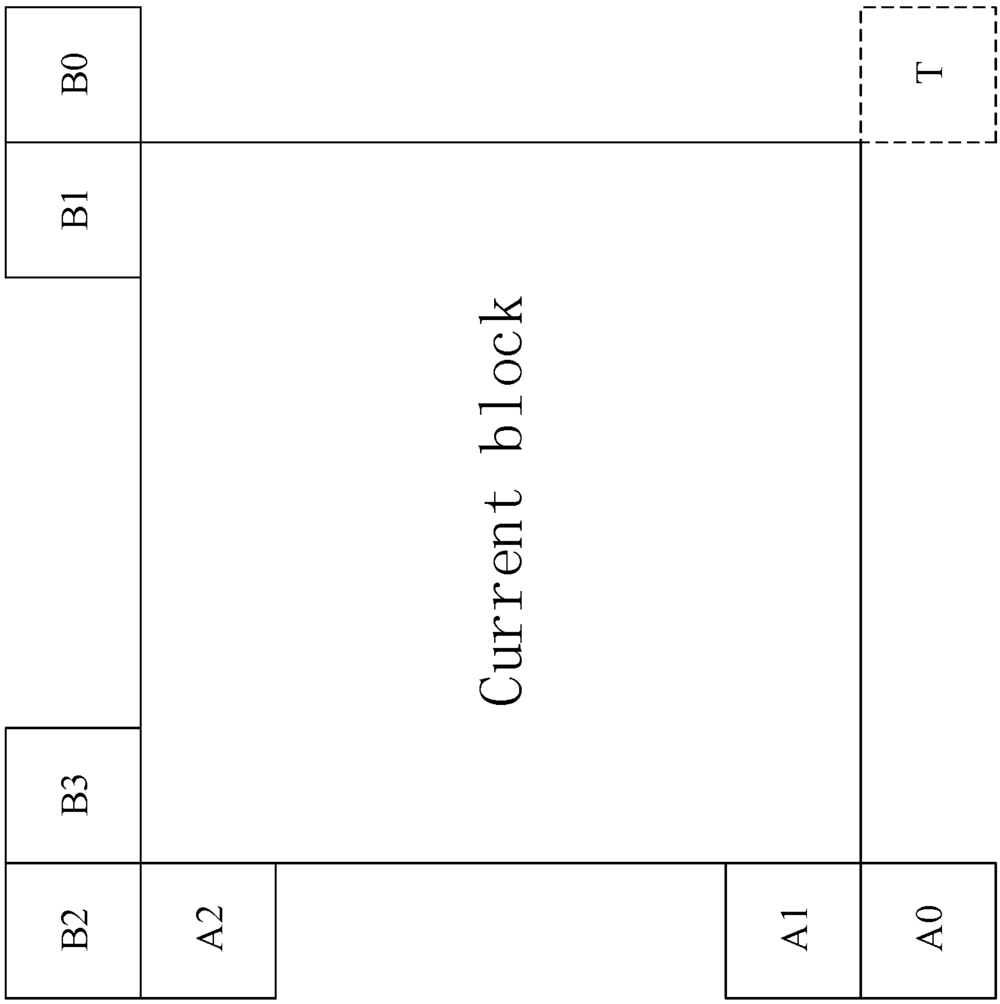

FIG. 13

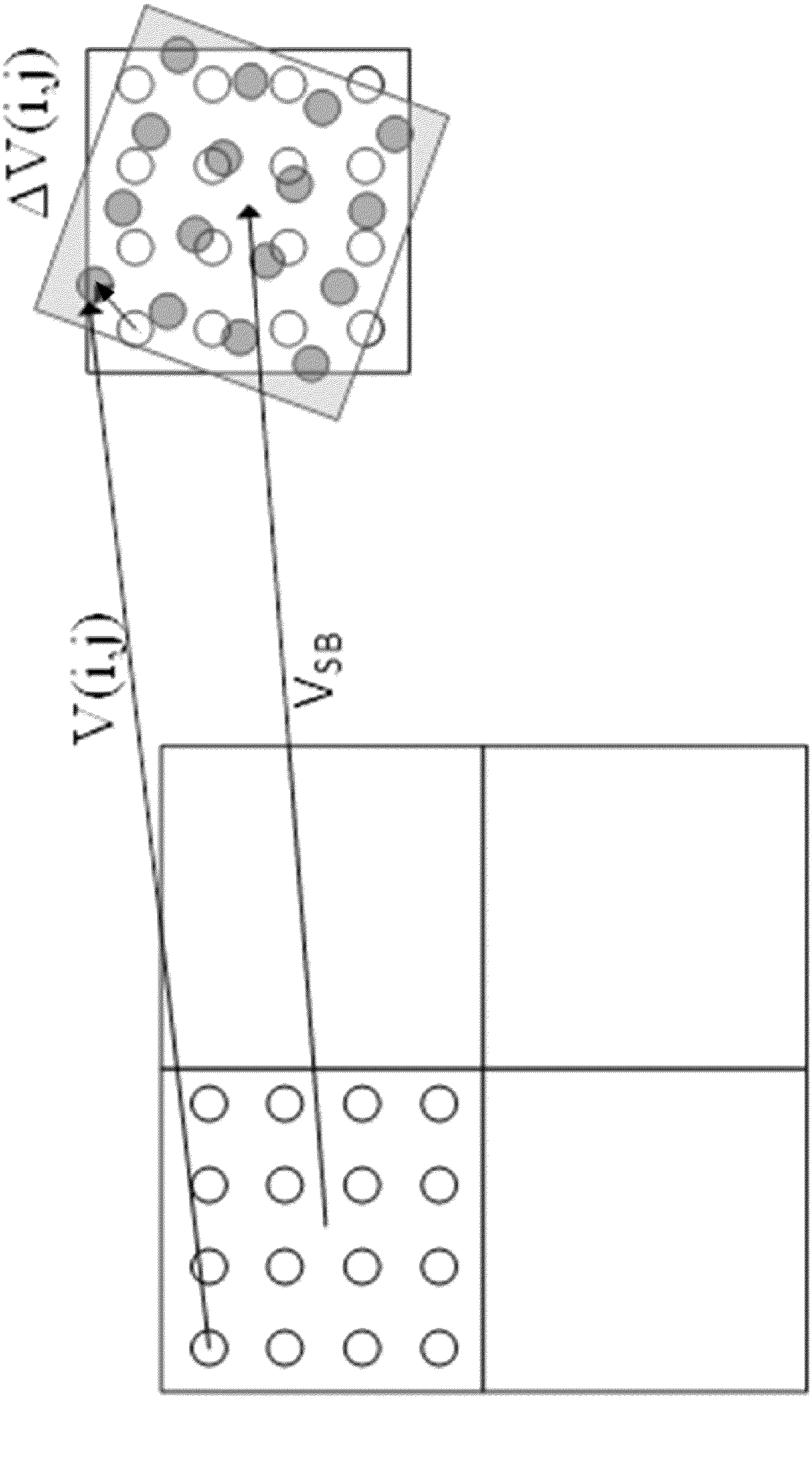

FIG. 14

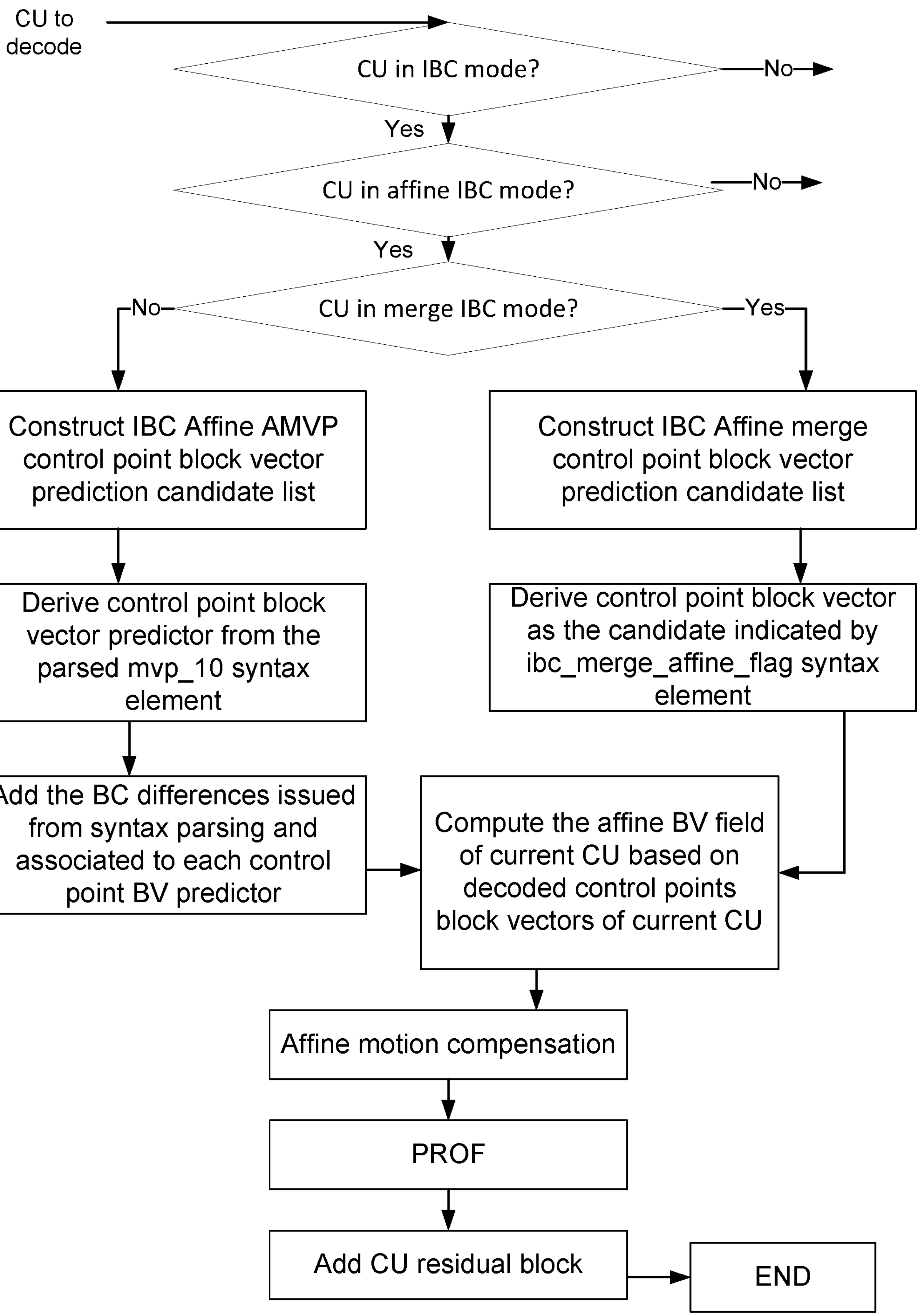
CU to decode
CU in IBC mode?
No
Yes
CU in affine IBC mode?
No
Yes
CU in merge IBC mode?
No
Yes
Construct IBC Affine AMVP control point block vector prediction candidate list
Construct IBC Affine merge control point block vector prediction candidate list
Derive control point block vector predictor from the parsed mvp_10 syntax element
Derive control point block vector as the candidate indicated by ibc_merge_affine_flag syntax element
Add the BC differences issued from syntax parsing and associated to each control point BV predictor
Compute the affine BV field of current CU based on decoded control points block vectors of current CU
Affine motion compensation
PROF
Add CU residual block
END

FIG. 15

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2024/066741

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04N19/105 H04N19/11 H04N19/139 H04N19/159 H04N19/176
H04N19/52 H04N19/54 H04N19/593 H04N19/70
ADD.

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

B. FIELDS SEARCHED

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

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

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

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 2023/046127 A1 (BEIJING BYTEDANCE NETWORK TECH CO LTD [CN]; BYTEDANCE INC [US]) 30 March 2023 (2023-03-30) abstract section "Summary", par. [004], page 1 section "2.4 Intra block copy (IBC)", pages 21-25 section "2.27. IBC Motion Candidates", pages 91-98 section "2.33.4 Prediction refinement with optical flow for affine mode", pages 111-113 section "2.35. IBC Mode Extension", subsection "4. Detailed Description", pages 126-134 par. [0110]-[0117], [121]-[0165], pages 134-141 -----	1-14

-/--

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

[x] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance;; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance;; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
5 August 2024	27/08/2024

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

1

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2021/321112 A1 (LIANG FAN [CN] ET AL) 14 October 2021 (2021-10-14) abstract paragraphs [0003] - [0007], [0085] - [0095], [0108] - [0113], [0118], [0141] - [0155] figures 6, 10 -----	1-14

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/EP2024/066741

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 2023046127 A1	30-03-2023	CN 118355659 A	16-07-2024
		WO 2023046127 A1	30-03-2023
US 2021321112 A1	14-10-2021	CN 113287309 A	20-08-2021
		EP 3902257 A1	27-10-2021
		US 2021321112 A1	14-10-2021
		WO 2020133115 A1	02-07-2020