

TEMPLATE-BASED SYNTAX ELEMENT PREDICTION

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of European Patent Application 21306335.7, filed September 27, 2021, the disclosure of which is incorporated herein by reference in its entirety.

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 related to the signaling of syntax elements. The syntax elements values for a block may be inferred, derived, and/or predicted from previously coded blocks (e.g., previously decoded blocks or previously encoded blocks), whose template samples (e.g., L-shaped pixels that surround the blocks) match the current block template samples.

[0004] In examples, a video decoder may determine whether a template-based coding mode may be enabled for a current block. Based on the determination of the template-based coding mode being enabled for the current block, a neighboring block (e.g., a decoded block) may be identified based on the template sample values of the current block. Value(s) of one or more syntax elements of the current block may be obtained based on the identified neighboring block. The current block may be decoded (e.g., reconstructed) based on the value(s) of the syntax element(s).

[0005] In examples, a video encoder may determine whether a template-based coding mode may be enabled for a current block. Based on the determination of the template-based coding mode being enabled for the current block, a neighboring block (e.g., a decoded block) may be identified based on template sample values of the current block. Value(s) of one or more syntax element(s) of the current block may be obtained based on the identified neighboring block. The current block may be encoded based on the value(s) of the syntax element(s).

[0006] In examples, a video encoder may determine whether to use a template-based coding mode for a current block. Based on the determination that template-based coding mode is used for the current block, signaling a syntax element for the current block may be excluded. The current block may be encoded based on

template-based coding mode. For example, a neighboring block (e.g., an encoded block) may be identified based on the template sample values of the current block. Value(s) of one or more syntax elements of the current block may be obtained based on the identified neighboring block (e.g., encoded block). The current block may be encoded based on the value(s) of the syntax element(s).

[0007] These examples may be performed by a device with a 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. Video data may include information representative of the template matching prediction mode. The video data may include a bitstream as described herein.

[0008] 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

[0009] FIG. 1A is a system diagram illustrating an example communications system in which one or more disclosed embodiments may be implemented.

[0010] 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.

[0011] 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.

[0012] 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.

[0013] FIG. 2 illustrates an example video encoder.

[0014] FIG. 3 illustrates an example video decoder.

- [0015] FIG. 4 illustrates an example of a system in which various aspects and examples may be implemented.
- [0016] FIG. 5 illustrates an example of template matching prediction (TMP).
- [0017] FIG. 6 illustrates an example of searching for matching templates for the current block inside of the decoded region.
- [0018] FIG. 7 illustrates an example of two templates of dimension 8 and 4 respectively.
- [0019] FIG. 8 illustrates an example flow chart for decoding a current block.
- [0020] FIG. 9 illustrates an example flow chart for encoding a current block.
- [0021] FIG. 10 illustrates an example flow chart for encoding a current block.

DETAILED DESCRIPTION

[0022] A more detailed understanding may be had from the following description, given by way of example in conjunction with the accompanying drawings.

[0023] 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.

[0024] 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.

[0025] 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.

[0026] 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.

[0027] 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).

[0028] 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 115/116/117 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).

[0029] 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).

[0030] 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).

[0031] 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., a eNB and a gNB).

[0032] 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.

[0033] 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.

[0034] 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.

[0035] 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.

[0036] 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.

[0037] 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.

[0038] 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.

[0039] 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.

[0040] 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.

[0041] 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.

[0042] 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).

[0043] 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.

[0044] 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.

[0045] 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.

[0046] 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 WTRU 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)).

[0047] 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.

[0048] 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.

[0049] 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.

[0050] 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.

[0051] The MME 162 may be connected to each of the eNode-Bs 162a, 162b, 162c 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.

[0052] 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.

[0053] 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.

[0054] 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.

[0055] 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.

[0056] In representative embodiments, the other network 112 may be a WLAN.

[0057] 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.

[0058] 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 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.

[0059] 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.

[0060] 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).

[0061] 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).

[0062] 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.

[0063] 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.

[0064] 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.

[0065] 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, 180b 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).

[0066] 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).

[0067] 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.

[0068] 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.

[0069] 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.

[0070] 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.

[0071] 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.

[0072] 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 184, 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.

[0073] 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.

[0074] 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.

[0075] 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 performing testing using over-the-air wireless communications.

[0076] 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 test 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.

[0077] 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.

[0078] The aspects described and contemplated in this application may be implemented in many different forms. FIGS. 5-7 described herein may provide some examples, but other examples are contemplated. The discussion of FIGS. 5-7 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.

[0079] In the present application, the terms “reconstructed” and “decoded” may be used interchangeably, the terms “pixel” and “sample” may be used interchangeably, the terms “image,” “picture” and “frame” may be used interchangeably.

[0080] 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 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.

[0081] 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 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.

[0082] Various numeric values are used in examples described 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.

[0083] FIG. 2 illustrates 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.

[0084] 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.

[0085] In the encoder 200, a picture is encoded by the encoder elements as described below. The picture to be encoded is partitioned (202) and processed in units of, for example, coding units (CUs). Each unit is encoded using, for example, either an intra or inter mode. When a unit is encoded in an intra mode, it performs intra prediction (260). In an inter mode, motion estimation (275) and compensation (270) are performed. The encoder decides (205) which one of the intra mode or inter mode to use for encoding the unit, and indicates the intra/inter decision by, for example, a prediction mode flag. Prediction residuals are calculated, for example, by subtracting (210) the predicted block from the original image block.

[0086] The prediction residuals are then transformed (225) and quantized (230). The quantized transform coefficients, as well as motion vectors and other syntax elements, are entropy coded (245) to output a bitstream. The encoder can skip the transform and apply quantization directly to the non-transformed residual signal. The encoder can bypass both transform and quantization, i.e., the residual is coded directly without the application of the transform or quantization processes.

[0087] The encoder decodes an encoded block to provide a reference for further predictions. The quantized transform coefficients are de-quantized (240) and inverse transformed (250) to decode prediction residuals. Combining (255) the decoded prediction residuals and the predicted block, an image block is reconstructed. In-loop filters (265) are applied to the reconstructed picture to perform, for example, deblocking/SAO (Sample Adaptive Offset) filtering to reduce encoding artifacts. The filtered image is stored at a reference picture buffer (280).

[0088] FIG. 3 illustrates an example of a video decoder. In example decoder 300, a bitstream is decoded by the decoder elements as described below. Video decoder 300 generally performs a decoding pass reciprocal to the encoding pass as described in FIG. 2. The encoder 200 also generally performs video decoding as part of encoding video data.

[0089] In particular, the input of the decoder includes a video bitstream, which may be generated by video encoder 200. The bitstream is first entropy decoded (330) to obtain transform coefficients, motion vectors, and other coded information. The picture partition information indicates how the picture is partitioned. The decoder may therefore divide (335) the picture according to the decoded picture partitioning information. The transform

coefficients are 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 is reconstructed. The predicted block may be obtained (370) from intra prediction (360) or motion-compensated prediction (i.e., inter prediction) (375). In-loop filters (365) are applied to the reconstructed image. The filtered image is stored at a reference picture buffer (380).

[0090] The decoded picture can further 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) or an inverse remapping performing the inverse of the remapping process performed in the pre-encoding processing (201). The post-decoding processing can 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.

[0091] FIG. 4 illustrates an example of a system in which various aspects and examples described herein may be implemented. System 400 may be embodied as a device including the various components described below and is configured to perform one or more of the aspects described in this document. Examples of such devices, 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. Elements of 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 system 400 are distributed across multiple ICs and/or discrete components. In various examples, the system 400 is 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 is configured to implement one or more of the aspects described in this document.

[0092] The system 400 includes at least one processor 410 configured to execute instructions loaded therein for implementing, for example, the various aspects described in this document. Processor 410 can include embedded memory, input output interface, and various other circuitries as known in the art. The system 400 includes at least one memory 420 (e.g., a volatile memory device, and/or a non-volatile memory device). System 400 includes a storage device 440, which can 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 can 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.

[0093] System 400 includes 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 can include its own processor and memory. The encoder/decoder module 430 represents module(s) that may be included in a device to perform the encoding and/or decoding functions. As is known, a device can include one or both of the encoding and decoding modules. Additionally, encoder/decoder module 430 may be implemented as a separate element of system 400 or may be incorporated within processor 410 as a combination of hardware and software as known to those skilled in the art.

[0094] Program code to be loaded onto processor 410 or encoder/decoder 430 to perform the various aspects described in this document may be stored in storage device 440 and subsequently loaded onto memory 420 for execution by processor 410. In accordance with various examples, one or more of processor 410, memory 420, storage device 440, and encoder/decoder module 430 can store one or more of various items during the performance of the processes described in this document. Such stored items can 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.

[0095] In some examples, memory inside of the processor 410 and/or the encoder/decoder module 430 is 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) is 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.

[0096] The input to the elements of system 400 may be provided through various input devices as indicated in block 445. Such input devices 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, include composite video.

[0097] In various examples, the input devices of block 445 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 includes 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 can 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 receives an RF signal transmitted over a wired (for example, cable) medium, and performs 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, remove some of these elements, and/or add other elements performing similar or different functions. Adding elements can 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 includes an antenna.

[0098] The USB and/or HDMI terminals can 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 processor 410 as necessary. Similarly, aspects of USB or HDMI interface processing may be implemented within separate interface ICs or within processor 410 as necessary. The demodulated, error corrected, and demultiplexed stream is provided to various processing elements, including, for example, processor 410, and encoder/decoder 430 operating in combination with the memory and storage elements to process the datastream as necessary for presentation on an output device.

[0099] 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 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.

[0100] The system 400 includes communication interface 450 that enables communication with other devices via communication channel 460. The communication interface 450 can include, but is not limited to, a transceiver configured to transmit and to receive data over communication channel 460. The communication interface 450 can 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.

[0101] Data is 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 is 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 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 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.

[0102] The system 400 can provide an output signal to various output devices, including a display 475, speakers 485, and other peripheral devices 495. The display 475 of various examples includes 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 can 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 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 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.

[0103] In various examples, control signals are communicated between the system 400 and the display 475, speakers 485, or other peripheral devices 495 using signaling such as AV.Link, Consumer Electronics Control (CEC), or other communications protocols 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 system 400 using the communications channel 460 via the communications interface 450. The display 475 and speakers 485 may be integrated in a single unit with the other components of system 400 in an electronic device such as, for example, a television. In various examples, the display interface 470 includes a display driver, such as, for example, a timing controller (T Con) chip.

[0104] The display 475 and speakers 485 can 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 speakers 485 are external components, the output signal may be provided via dedicated output connections, including, for example, HDMI ports, USB ports, or COMP outputs.

[0105] 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 can encompass one or more of microprocessors, general purpose computers, special purpose computers, and processors based on a multi-core architecture, as non-limiting examples.

[0106] 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, include processes performed by a decoder of various implementations described in this application, for example, determining whether a template-based coding mode is enabled for a current block, based on the determination of the template-based coding mode being enabled for the current block, identifying at least one decoded block based on template sample values of the identified decoded block and the current block, obtaining a value of at least one syntax element of the current block based on the identified at least one decoded block, and reconstructing the current block using the value of the at least one syntax element, etc.

[0107] 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.

[0108] 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, include processes performed by an encoder of various implementations described in this application, for example, determining whether to use a template-based coding mode for a current block, based on the determination of template-based coding mode being enabled for the current block, excluding signaling of at least one syntax element for the current block, and encoding the current block based on template-based coding mode, etc.

[0109] 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.

[0110] Note that syntax elements as used herein, for example, coding syntax on template matching prediction, including but not limited to, `cu_template_based_coding_enabled_flag`, `cu_intra_template_based_coding_enabled_flag`, `cu_inter_template_based_coding_enabled_flag`, `cu_transform_template_based_coding_enabled_flag` are descriptive terms. As such, they do not preclude the use of other syntax element names.

[0111] 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.

[0112] 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.

[0113] 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.

[0114] 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.

[0115] 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.

[0116] 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.

[0117] It is to be appreciated that the use of any of the following "I", "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.

[0118] 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 is 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.

[0119] 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 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.

[0120] 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.

[0121] The syntax elements values may be predicted from previously coded blocks whose template (e.g., L-shaped samples surrounding the blocks) match the current block template. The examples described herein may increase coding gain and/or decrease signaling of syntax elements.

[0122] For example, an encoder may determine whether to use a template-based coding mode for the current block. Based on the determination to use template-based coding mode for the current block, the encoder may bypass signaling of at least one syntax element for the current block. The current block may be encoded based on template-based coding mode. Based on the determination that template-based coding mode is not used for the current block, the at least one syntax element for the current block may be included in the bitstream.

[0123] These examples may be performed by a device with at least one processor. The device may include an encoder and/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. These examples may be performed by a bitstream comprising information representative of template matching prediction mode.

[0124] FIG. 5 illustrates an example of template matching prediction (TMP). TMP may be an intra prediction mode that may copy a prediction block (e.g., the best prediction block) from the reconstructed part of the current frame, whose template (e.g., L-shaped template) matches the current template. For a predefined search range, the encoder may search for the most similar template to the current template in an encoded part of the current frame and may use the corresponding block as a prediction block. The encoder may indicate (e.g., signal in the bitstream) the usage of the template matching prediction mode, and the prediction operation (e.g., the corresponding prediction operation) may be performed at the decoder side. The blocks associated with the templates (e.g., target blocks) may be used to generate the prediction signal. In examples, the prediction signal may be generated by averaging the templates. In examples, the prediction signal may be generated by considering the block that has minimum template difference. TMP may be performed in

conjunction with intra sub-partitions (ISP), matrix-weighted intra prediction (MIP), and/or multiple reference line (MRL) intra prediction, may have interaction with transform tools (e.g., multi transform selection (MTS) and/or low-frequency non-separable transform (LFNST)), and/or may have interaction with combined inter and intra prediction. An indication, such as coding unit (CU) flag, may be signaled to indicate the usage of TMP. This indication may be signaled at different levels (e.g., at the sub-CU level, at the transform unit level, at the prediction unit level, at the slice level) in the codec designed.

[0125] Syntax elements (e.g., several syntax elements) may be indicated to the decoder to perform the inverse process and reconstruct the pixels from the bitstream. In examples, at the CU level, various indications (e.g., flags) may be signaled to indicate the prediction type, the transform type and other tools being enabled or disabled. An example CU signaling is indicated below (syntax elements are in **bold**):

coding_unit(x0, y0, cbWidth, cbHeight, cqtDepth, treeType, modeType) {	Descriptor
...	
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) {	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA) {	
if(pred_mode_plt_flag)	
palette_coding(x0, y0, cbWidth, cbHeight, treeType)	
else {	
if(sps_bdpcm_enabled_flag && cbWidth <= MaxTsSize && cbHeight <= MaxTsSize)	
intra_bdpcm_luma_flag	ac(v)
if(intra_bdpcm_luma_flag)	
intra_bdpcm_luma_dir_flag	ac(v)
else {	
if(sps_mip_enabled_flag)	
intra_mip_flag	ac(v)
if(intra_mip_flag) {	
intra_mip_transposed_flag [x0][y0]	ac(v)
intra_mip_mode [x0][y0]	ac(v)
} else {	
...	

[0126] The number of syntax elements coded at the block level (e.g., CU, PU and/or TU) may be proportional to the number tools used. For example, each tool may have a flag to indicate its usage. Many tools may have their parameters signaled if they are used. This may lead to excessive signaling of flags for each block, which may increase the overhead and reduce the overall gain.

[0127] In examples, the syntax elements of a coding block may be inferred, derived, and/or predicted from another decoded block if their templates match (e.g., the template sample values are the same or substantially similar). The syntax elements or a subset of syntax elements for the coding block may be obtained from the decoded block with matching templates.

[0128] Whether to use a template-based coding mode for a current block may be determined. Based on the determination of the template-based coding mode being enabled for the current block, certain syntax element(s) for the current block may be excluded from the bitstream. An indication may be signaled at the beginning of the block syntax structure to indicate the usage of the template-based coding mode for the current block. In examples, at the CU level, a flag (e.g., cu_template_based_coding_enabled_flag) may be signaled to indicate whether template-based coding mode is used to code the current block. If this flag is equal to one, all (or some) of the other syntax elements may be skipped from signaling and inferred from the matching block. In examples, flags for intra prediction modes (e.g., intra_mip_flag and intra_subpartitions_mode) may be skipped. Based on the determination of the template-based coding mode being disabled for the current block, the syntax element(s) for the current block may be signaled in the bitstream. If a template-based coding enabled

indication described herein is absent in the bitstream, its value may be inferred to indicate that template-based coding is disabled. If a template-based coding enabled indication indicates that template-based coding is enabled, the syntax template deduction described herein may be used.

[0129] A decoder may determine whether template-based coding mode is enabled for the current block. The determination may be based on a template-based coding enabled indication such as `cu_template_based_coding_enabled_flag`. Based on the determination of the template-based coding mode being enabled for the current block, a neighboring block (e.g., a decoded block) may be identified via template matching. The neighboring block (e.g., the decoded block) may be identified based on template sample values of the identified neighboring block (e.g., decoded block) and template sample values of the current block. For example, the neighboring block may be identified by having matching template samples (e.g., identical template sample values, closest template sample values, or similar template sample values) as the template samples of the current block. Values of at least one syntax element of the current block may be obtained based on the identified neighboring block (e.g., decoded block). For example, values of syntax element(s) such as flags for intra prediction modes (e.g., `intra_mip_flag` and `intra_subpartitions_mode`) for the current block may be inferred, derived, and/or predicted from the corresponding syntax element(s) of the identified neighboring block (e.g., decoded block). The current block may be decoded (e.g., reconstructed) using the values of the syntax elements obtained via template matching.

[0130] An encoder may determine whether template-based coding mode is enabled for the current block. The determination may be based on rate-distortion optimization. A template-based coding enablement indication, such as `cu_template_based_coding_enabled_flag`, may be included in the video data to indicate whether template-based coding mode is enabled (e.g., for a coding block, for multiple coding blocks, for a sub-block, and/or the like). Based on the determination of the template-based coding mode being enabled for the current block, one or more syntax elements for the current block may be excluded from the video data (e.g., signaling of the syntax element(s) may be bypassed). The current block may be encoded using the values of the syntax elements obtained via template matching.

[0131] An example of CU signaling is shown in the table below:

	Descriptor
<code>coding_unit(x0, y0, cbWidth, cbHeight, cqtDepth, treeType, modeType) {</code>	
<code>if(sh_slice_type == I && (cbWidth > 64 cbHeight > 64))</code>	
<code>modeType = MODE_TYPE_INTRA</code>	
<code>chType = treeType == DUAL_TREE_CHROMA ? 1 : 0</code>	
<code>cu_template_based_coding_enabled_flag</code>	<code>ae(v)</code>

if(cu_template_based_coding_enabled_flag == false)	
{	
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) {	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA) {	
if(pred_mode_plt_flag)	
palette_coding(x0, y0, cbWidth, cbHeight, treeType)	
else {	
if(sps_bdpcm_enabled_flag && cbWidth <= MaxTsSize && cbHeight <= MaxTsSize)	
intra_bdpcm_luma_flag	ae(v)
if(intra_bdpcm_luma_flag)	
intra_bdpcm_luma_dir_flag	ae(v)
else {	
if(sps_mip_enabled_flag)	
intra_mip_flag	ae(v)
if(intra_mip_flag) {	
intra_mip_transposed_flag [x0][y0]	ae(v)
intra_mip_mode [x0][y0]	ae(v)
} else {	
if(sps_mrl_enabled_flag && ((y0 % CtbSizeY) > 0))	

intra_luma_ref_idx	ac(v)
if(sps_isp_enabled_flag && intra_luma_ref_idx == 0 && (cbWidth <= MaxTbSizeY && cbHeight <= MaxTbSizeY) && (cbWidth * cbHeight > MinTbSizeY * MinTbSizeY) && !cu_act_enabled_flag)	
intra_subpartitions_mode_flag	ac(v)
if(intra_subpartitions_mode_flag == 1)	
...	
}	

[0132] FIG. 6 illustrates an example of searching for matching templates for the current block (e.g., inside of the decoded part). In examples, the signaling overhead may be reduced by predicting the syntax elements from neighboring blocks (e.g., previously decoded blocks within the decoded part or previously encoded blocks in the encoded part). Neighboring block(s) (e.g., decoded block(s) or encoded blocks) whose template matches the current one may be searched for. In examples, the neighboring block(s) (e.g., decoded block(s)) may be identified based on the template sample values of the neighboring block(s) (e.g., decoded block(s)) and the template sample values of the current block. The template sample values of the identified neighboring block(s) (e.g., decoded block(s)) may match the template sample values of the current block. For example, the matching template sample values of the neighboring block(s) (e.g., decoded block(s)) may be identical template sample values, the closest template sample values, or similar template sample values as the template samples of the current block. The value(s) of at least one syntax element of the current block may be obtained based on the identified at least one neighboring block (e.g., decoded block). This may be performed on both the encoder and decoder side.

[0133] To find the matching block, searching in all the decoding blocks may not be needed. There may be a table registering information of decoded blocks. In examples, a video decoding device and/or a video encoding device may register the syntax element values of neighboring blocks into the table. The table may be a table of size N whose entries may be the template of each neighboring block and all (or some) of its syntax elements. This is shown in the table below, where S1, S2, ... are the syntax elements that may be used for template-based coding:

Template	S1	S2	S3	S4	S5	S6	...
X	X	X	X	X	X	X	...
X	X	X	X	X	X	X	...

...	...	...	...	...	...	...	...
-----	-----	-----	-----	-----	-----	-----	-----

[0134] Based on the table, the video decoding device and/or the video encoding device may obtain a syntax element value of identified neighboring block that corresponds to the syntax element of the current block. The value of the syntax element of the current block may be obtained based on the syntax element value of the identified neighboring block. In examples, the video decoding device may set the value of the syntax element of the current block to the syntax element value of the identified neighboring block. In examples, the video decoding device may predict the value of the syntax element of the current block based on the syntax element value of the identified neighboring block. In examples, the video decoding device may copy the value of the syntax element of the current block to the syntax element value of the identified neighboring block.

[0135] The table may be updated with new entries when finishing reconstruction of a block or encoding a block. The table may be updated with distinctive templates (e.g., to avoid having redundant information inside the table). To have distinctive templates, the distance between templates may be measured and the information from blocks with high distance may be inserted in the table. The absolute difference or square difference may be used as a distance measure.

[0136] In examples, a single candidate may be used. A best neighboring block (e.g., best decoded block) may be identified based on the template sample values of the best identified neighboring block (e.g., best decoded block) and the template sample values of the current block. The value(s) of at least one syntax element of the current block may be obtained based on best neighboring block (e.g., best decoded block). The values of the syntax element(s) of the best neighboring block (e.g., decoded block) may be used to decode (e.g., reconstruct) the current block.

[0137] A best neighboring block (e.g., best encoded block) may be identified based on the template sample values of the best identified neighboring block (e.g., best encoded block) and the template sample values of the current block. The value(s) of at least one syntax element of the current block may be obtained based on best neighboring block (e.g., best encoded block). The values of the syntax element(s) of the best neighboring block may be used to encode the current block.

[0138] In examples, multiple candidates may be used. Multiple neighboring blocks (e.g., decoded blocks) may be identified based on the template sample values of the neighboring blocks (e.g., decoded blocks) and template sample values of the current block. The values of at least one syntax element of the current block may be obtained based on the identified neighboring blocks (e.g., decoded blocks). The values of the at least

one syntax element may be used to decode (e.g., reconstruct) the current block. In examples, a first subset of syntax elements of one or more neighboring blocks (e.g., decoded blocks) may be used to decode the current block, and a second subset syntax elements of one or more other neighboring blocks (e.g., decoded blocks) may be used to decode the current block. The encoder may indicate the block or blocks used for prediction to the decoder. The table may be sorted according to the distance to current template. The encoder may test up to M candidates of the table to find the best candidate and may signal the index of the best neighboring block (e.g., decoded block).

[0139] In examples, template-based coding mode may be enabled for a class of syntax elements for the current block. Based on the determination of the template-based coding mode being enabled for the class of syntax elements for the current block, values of the class of syntax elements of the current block may be obtained based on corresponding syntax element values of the identified neighboring block(s) (e.g., decoded block(s)). The current block may be decoded (e.g., reconstructed) based on the values of the class of syntax elements of the current block.

[0140] In examples, a video encoding device may enable template-based coding mode for a class of syntax elements for the current block. Based on the determination of the template-based coding mode being enabled for the class of syntax elements for the current block, the class of syntax elements of the current block may be excluded from video data (e.g., signaling of the syntax elements in the class may be bypassed for the current block). The current block may be encoded based on template-based coding mode. Whether to code the current block using the template-based coding mode may be determined based on rate-distortion optimization.

[0141] The class of syntax elements may include at least one of intra prediction syntax elements, inter prediction syntax elements, or transform syntax elements. In examples, the class of syntax elements may include one of intra prediction syntax elements, inter prediction syntax elements, or transform syntax elements. In examples, the class of syntax elements may include two of intra prediction syntax elements, inter prediction syntax elements, or transform syntax elements. In examples, the class of syntax elements may include all three of intra prediction syntax elements, inter prediction syntax elements, or transform syntax elements.

[0142] One or more flags may be signaled to indicate the usage of template-based code mode for intra prediction syntax elements, inter prediction syntax elements, and/or transform syntax elements. The usage of specific tools, such as intra subpartitioning, or subblock transform, may be indicated. An example of syntax is indicated below:

	Descriptor
coding_unit(x0, y0, cbWidth, cbHeight, cqtDepth, treeType, modeType) {	
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(CuPredMode[chType][x0][y0] == MODE_INTRA)	
cu_intra_template_based_coding_enabled_flag	ae(v)
if(CuPredMode[chType][x0][y0] == MODE_INTER)	
cu_inter_template_based_coding_enabled_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 && cu_intra_template_based_coding_enabled_flag == false && cu_inter_template_based_coding_enabled_flag == false)	
pred_mode_ibc_flag	ae(v)
} else	
cu_intra_template_based_coding_enabled_flag	ae(v)
if(CuPredMode[chType][x0][y0] == MODE_INTRA && cu_intra_template_based_coding_enabled_flag == false && 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) {	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA) {	
if(pred_mode_plt_flag)	
palette_coding(x0, y0, cbWidth, cbHeight, treeType)	
else {	
if(sps_bdpcm_enabled_flag && cbWidth <= MaxTsSize && cbHeight <= MaxTsSize)	
intra_bdpcm_luma_flag	ae(v)

if(intra_bdpcm_luma_flag)	
intra_bdpcm_luma_dir_flag	ae(v)
else {	
if(sps_mip_enabled_flag)	
intra_mip_flag	ae(v)
if(intra_mip_flag) {	
intra_mip_transposed_flag [x0][y0]	ae(v)
intra_mip_mode [x0][y0]	ae(v)
} else {	
if(sps_mrl_enabled_flag && ((y0 % CtbSizeY) > 0))	
intra_luma_ref_idx	ae(v)
if(sps_ism_enabled_flag && intra_luma_ref_idx == 0 && (cbWidth <= MaxTbSizeY && cbHeight <= MaxTbSizeY) && (cbWidth * cbHeight > MinTbSizeY * MinTbSizeY) && !cu_act_enabled_flag)	
intra_subpartitions_mode_flag	ae(v)
if(intra_subpartitions_mode_flag == 1)	
intra_subpartitions_split_flag	ae(v)
if(intra_luma_ref_idx == 0)	
intra_luma_mpm_flag [x0][y0]	ae(v)
if(intra_luma_mpm_flag[x0][y0]) {	
if(intra_luma_ref_idx == 0)	
intra_luma_not_planar_flag [x0][y0]	ae(v)
if(intra_luma_not_planar_flag[x0][y0])	
intra_luma_mpm_idx [x0][y0]	ae(v)
} else	
intra_luma_mpm_remainder [x0][y0]	ae(v)
}	
}	
}	
}	
if((treeType == SINGLE_TREE treeType == DUAL_TREE_CHROMA) && sps_chroma_format_idc != 0) {	
if(pred_mode_plt_flag && treeType == DUAL_TREE_CHROMA)	
palette_coding(x0, y0, cbWidth / SubWidthC, cbHeight / SubHeightC, treeType)	
else if(!pred_mode_plt_flag) {	
if(!cu_act_enabled_flag) {	
if(cbWidth / SubWidthC <= MaxTsSize && cbHeight / SubHeightC <= MaxTsSize && sps_bdpcm_enabled_flag)	
intra_bdpcm_chroma_flag	ae(v)
if(intra_bdpcm_chroma_flag)	
intra_bdpcm_chroma_dir_flag	ae(v)
else {	

if(CclmEnabled)	
cclm_mode_flag	ae(v)
if(cclm_mode_flag)	
cclm_mode_idx	ae(v)
else	
intra_chroma_pred_mode	ae(v)
}	
}	
}	
}	
} else if(treeType != DUAL_TREE_CHROMA && cu_inter_template_based_coding_enabled_flag == false) { /* 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) {	
mvd_coding(x0, y0, 0, 0)	
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]	ac(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]	ac(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(MotionModelIdx[x0][y0] > 0)	
mvd_coding(x0, y0, 1, 1)	
if(MotionModelIdx[x0][y0] > 1)	
mvd_coding(x0, y0, 1, 2)	
}	
mvp_l1_flag [x0][y0]	ac(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]	ac(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_10_flag[ref_idx_10 [x0][y0]] == 0 && luma_weight_11_flag[ref_idx_11 [x0][y0]] == 0 && chroma_weight_10_flag[ref_idx_10 [x0][y0]] == 0 && chroma_weight_11_flag[ref_idx_11 [x0][y0]] == 0 && cbWidth * cbHeight >= 256)	
bcw_idx [x0][y0]	ae(v)
}	
}	
if(CuPredMode[chType][x0][y0] != MODE_INTRA && !pred_mode_plt_flag && general_merge_flag[x0][y0] == 0)	
cu_coded_flag	ae(v)
if(cu_coded_flag) {	
cu_transform_template_based_coding_enabled_flag	ae(v)
if(cu_transform_template_based_coding_enabled_flag == false) {	
if(CuPredMode[chType][x0][y0] == MODE_INTER && sps_sbt_enabled_flag && !ciip_flag[x0][y0] && cbWidth <= MaxTbSizeY && cbHeight <= MaxTbSizeY) {	
allowSbtVerH = cbWidth >= 8	
allowSbtVerQ = cbWidth >= 16	
allowSbtHorH = cbHeight >= 8	
allowSbtHorQ = cbHeight >= 16	
if(allowSbtVerH allowSbtHorH)	
cu_sbt_flag	ae(v)
if(cu_sbt_flag) {	
if((allowSbtVerH allowSbtHorH) && (allowSbtVerQ allowSbtHorQ))	
cu_sbt_quad_flag	ae(v)
if((cu_sbt_quad_flag && allowSbtVerQ && allowSbtHorQ) (!cu_sbt_quad_flag && allowSbtVerH && allowSbtHorH))	
cu_sbt_horizontal_flag	ae(v)
cu_sbt_pos_flag	ae(v)
}	
}	
if(sps_act_enabled_flag && CuPredMode[chType][x0][y0] != MODE_INTRA && treeType == SINGLE_TREE)	
cu_act_enabled_flag	ae(v)
LfstDcOnly = 1	
LfstZeroOutSigCoeffFlag = 1	
MtsDcOnly = 1	
MtsZeroOutSigCoeffFlag = 1	
transform_tree(x0, y0, cbWidth, cbHeight, treeType, chType)	

lfnstWidth = (treeType == DUAL_TREE_CHROMA) ? cbWidth / SubWidthC : ((IntraSubPartitionsSplitType == ISP_VER_SPLIT) ? cbWidth / NumIntraSubPartitions : cbWidth)	
lfnstHeight = (treeType == DUAL_TREE_CHROMA) ? cbHeight / SubHeightC : ((IntraSubPartitionsSplitType == ISP_HOR_SPLIT) ? cbHeight / NumIntraSubPartitions : cbHeight)	
lfnstNotTsFlag = (treeType == DUAL_TREE_CHROMA !tu_y_coded_flag[x0][y0] transform_skip_flag[x0][y0][0] == 0) && (treeType == DUAL_TREE_LUMA ((!tu_cb_coded_flag[x0][y0] transform_skip_flag[x0][y0][1] == 0) && (!tu_cr_coded_flag[x0][y0] transform_skip_flag[x0][y0][2] == 0)))	
if(Min(lfstWidth, lfstHeight) >= 4 && sps_lfst_enabled_flag == 1 && CuPredMode[chType][x0][y0] == MODE_INTRA && lfstNotTsFlag == 1 && (treeType == DUAL_TREE_CHROMA !IntraMipFlag[x0][y0] Min(lfstWidth, lfstHeight) >= 16) && Max(cbWidth, cbHeight) <= MaxTbSizeY) {	
if((IntraSubPartitionsSplitType != ISP_NO_SPLIT LfstDcOnly == 0) && LfstZeroOutSigCoeffFlag == 1)	
lfst_idx	ae(v)
}	
if(treeType != DUAL_TREE_CHROMA && lfst_idx == 0 && transform_skip_flag[x0][y0][0] == 0 && Max(cbWidth, cbHeight) <= 32 && IntraSubPartitionsSplitType == ISP_NO_SPLIT && cu_sbt_flag == 0 && MtsZeroOutSigCoeffFlag == 1 && MtsDcOnly == 0) {	
if(((CuPredMode[chType][x0][y0] == MODE_INTER && sps_explicit_mts_inter_enabled_flag) (CuPredMode[chType][x0][y0] == MODE_INTRA && sps_explicit_mts_intra_enabled_flag)))	
mts_idx	ae(v)
}	
}	
}	
}	

[0143] In the above example, template-based coding enabled indications may be signaled to indicate whether intra, inter and/or transform information are predicted (e.g., copied, inferred, derived) from the identified neighboring block (e.g., decoded block). Based on the template-based coding enabled indication(s), intra, inter and/or transform information may be predicted (e.g., copied, inferred, derived) from the identified neighboring block (e.g., decoded block).

[0144] For example, an intra template-based coding enabled indication, such as cu_intra_template_based_coding_enabled_flag shown in the example coding unit syntax table above, may

indicate whether intra prediction information of the current block is to be obtained (e.g., predicated, copied, inferred, derived) based on the intra prediction information associated with a neighboring block identified via template matching. For example, an inter template-based coding enabled indication, such as `cu_inter_template_based_coding_enabled_flag` shown in the example syntax table above, may indicate whether inter prediction information of the current block is to be obtained (e.g., predicted, copied, inferred, derived) based on the inter prediction information associated with a neighboring block identified via template matching. For example, a transform template-based coding enabled indication, such as the `cu_transform_template_based_coding_enabled_flag` shown in the example syntax table above, may indicate whether transform information of the current block is to be obtained (e.g., predicated copied, inferred, derived) based on the transform information associated with the identified neighboring block a neighboring block identified via template matching.

[0145] Although the example indications shown above may be signaled at the CU level, the indications may be signaled at other levels, such as at a slice level, at a tile level, at a subblock level, etc.

[0146] In examples, a subset of the prediction or transform information or values may be inferred from the identified neighboring block(s) (e.g., via template deduction). In examples, not all syntax values may be inferred based on the neighboring block identified via template matching. The values of the syntax elements from the identified neighboring block may be used to initialize the context of the entropy coding of the syntax element. In examples, the `cu_skip_flag` entropy coding model may be inferred from the `cu_skip_flag` values of the top and left CU of the current block. In examples, the entropy coding model may be inferred from the `cu_skip_flag` value of the identified neighboring block.

[0147] If searching for a similar template, the neighboring blocks (e.g., decoded blocks) and the current block may or may not have the same dimensions when computing the distance measure between the templates. The dimensionality of the neighboring blocks (e.g., decoded blocks) may be compared to the dimensionality of the current block. If the dimensionality of the neighboring block is larger than the current block, a template of the neighboring block may be subsampled to equal a template size of the current block. The neighboring block may be identified based on template sample values in the subsampled template of the neighboring block and the template sample values of the current block. If the dimensionality of the current block is larger than the neighboring block, a template of the current block may be subsampled to equal a template size of the neighboring block. The neighboring block may be identified based on template sample values in the subsampled template of the current block. Distance measures with different template sizes may be used.

[0148] FIG. 7 illustrates an example of two templates of dimension 8 and 4 respectively. For two blocks of sizes $N \times M$ and $K \times L$, two distance measures may be provided between the upper templates of sizes N and K , and between the left templates of sizes M and L . For each of the two templates to compare, the larger dimension may be 2^n times the smaller dimension. This is because of the split type of quad, binary, and ternary tree. Therefore, to compare two templates, the larger template may be subsampled to same dimension as the small one.

[0149] In some examples, the templates (e.g., the template of the current block and the template of the neighboring block) may both be subsampled. For example, the templates may be subsampled to the minimum CU size (e.g., 4 pixels) and the templates may be compared regardless of their dimensions. This may simplify the design and reduce the information stored in the template table.

[0150] In some examples, the common part may be compared (e.g., the part corresponding to the smallest template may be taken into account in the computation). The number of samples used to compare the blocks may be used to weigh the template distance function.

[0151] In examples, template search and template table may be restricted to the current CTU or CTU line. This may reduce the overall complexity and improve the coding speed by allowing parallel processing, which may allow decoding CTU lines in parallel where no dependencies are between them. In examples, template search and template table may not be restricted to the current CTU or CTU line.

[0152] FIG. 8 illustrates an example flow chart 800 for decoding a current block. At 802, it may be determined whether a template-based coding mode is enabled for a current block. At 804, based on the determination of the template-based coding mode being enabled for the current block, a neighboring block may be identified based on template sample values of the current block. At 806, a value of a syntax element of the current block may be obtained based on the identified neighboring block. At 808, the current block may be decoded based on the value of the syntax element.

[0153] FIG. 9 illustrates an example flow chart 900 for encoding a current block. At 902, it may be determined whether a template-based coding mode is enabled for a current block. At 904, based on the determination of the template-based coding mode being enabled for the current block, a neighboring block may be identified based on template sample values of the current block. At 906, a value of a syntax element of the current block may be obtained based on the identified neighboring block. At 908, the current block may be encoded based on the value of the syntax element.

[0154] FIG. 10 illustrates an example flow chart 1000 for encoding a current block. At 1002, it may be determined whether a template-based coding mode is enabled for a current block. At 1004, based on the determination of the template-based coding mode being enabled for the current block, a signaling of a syntax element may be excluded for the current block. At 1006, the current block may be encoded based on the template-based coding mode.

[0155] 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

What is Claimed:

1. A device for video decoding, comprising:

a processor configured to:

determine whether a template-based coding mode is enabled for a current block;
based on the determination of the template-based coding mode being enabled for the current block, identify a neighboring block based on a plurality of template sample values of the current block;
obtain a value of a syntax element of the current block based on the identified neighboring block; and
decode the current block based on the value of the syntax element.

2. The device of claim 1, wherein the processor is further configured to:

register syntax element values of a plurality of neighboring blocks into a table; and
obtain, based on the table, a syntax element value of the identified neighboring block that corresponds to the syntax element of the current block, wherein the value of the syntax element of the current block is obtained based on the syntax element value of the identified neighboring block.

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

obtain a syntax element value of the identified neighboring block that corresponds to the syntax element of the current block; and
set the value of the syntax element of the current block to the syntax element value of the identified neighboring block.

4. The device of claim 1, wherein the processor is further configured to:

identify a plurality of neighboring blocks based on a plurality of template sample values of the plurality of neighboring blocks and the plurality of template sample values of the current block; and
obtain the value of the syntax element of the current block based on the identified plurality of neighboring blocks.

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

compare a dimensionality of the neighboring block to a dimensionality of the current block;

based on the dimensionality of the neighboring block being larger than the current block, subsample a template of the neighboring block to equal a template size of the current block, wherein the neighboring block is identified based on a plurality of template sample values in the subsampled template of the neighboring block and the plurality of template sample values of the current block; and

based on the dimensionality of the current block being larger than the neighboring block, subsample a template of the current block to equal a template size of the neighboring block, wherein the neighboring block is identified based on a plurality of template sample values in the subsampled template of the current block.

6. The device of claim 1, wherein the template-based coding mode is enabled for a class of syntax elements for the current block, and wherein the processor is further configured to:

based on the determination of the template-based coding mode being enabled for the class of syntax elements for the current block, obtain values of the class of syntax elements of the current block based on corresponding syntax element values of the identified neighboring block; and

reconstruct the current block using the values of the class of syntax elements of the current block.

7. A method for video decoding, comprising:

determining whether a template-based coding mode is enabled for a current block;

based on the determination of the template-based coding mode being enabled for the current block, identifying a neighboring block based on a plurality of template sample values of the current block;

obtaining a value of a syntax element of the current block based on the identified neighboring block;

and

decoding the current block based on the value of the syntax element.

8. The method of claim 7, further comprising:

registering syntax element values of a plurality of neighboring blocks into a table; and

obtaining, based on the table, a syntax element value of the identified neighboring block that corresponds to the syntax element of the current block, wherein the value of the syntax element of the current block is obtained based on the syntax element value of the identified neighboring block.

9. The method of claim 7, further comprising:

obtaining a syntax element value of the identified neighboring block that corresponds to the syntax element of the current block; and

setting the value of the syntax element of the current block to the syntax element value of the identified neighboring block.

10. The method of claim 7, further comprising:

identifying a plurality of neighboring blocks based on a plurality of template sample values of the plurality of neighboring blocks and the plurality of template sample values of the current block; and

obtaining the value of the syntax element of the current block based on the identified plurality of neighboring blocks.

11. The method of claim 7, further comprising:

comparing a dimensionality of the neighboring block to a dimensionality of the current block;

based on the dimensionality of the neighboring block being larger than the current block, subsampling a template of the neighboring block to equal a template size of the current block, wherein the neighboring block is identified based on a plurality of template sample values in the subsampled template of the neighboring block and the plurality of template sample values of the current block; and

based on the dimensionality of the current block being larger than the neighboring block, subsampling a template of the current block to equal a template size of the neighboring block, wherein the neighboring block is identified based on a plurality of template sample values in the subsampled template of the current block.

12. The method of claim 7, wherein the template-based coding mode is enabled for a class of syntax elements for the current block, further comprising:

based on the determination of the template-based coding mode being enabled for the class of syntax elements for the current block, obtaining values of the class of syntax elements of the current block based on corresponding syntax element values of the identified neighboring block; and

reconstructing the current block using the values of the class of syntax elements of the current block.

13. The device of claim 6 or the method of claim 12, wherein the class of syntax elements comprise intra prediction syntax elements.

14. The device of claim 6 or the method of claim 12, wherein the class of syntax elements comprise inter prediction syntax elements.
15. The device of claim 6 or the method of claim 12, wherein the class of syntax elements comprise transform syntax elements.
16. A computer program product which is stored on a non-transitory computer readable medium and comprises program code instructions for implementing the steps of a method according to at least one of claims 7 to 15 when executed by at least one processor.
17. A computer program comprising program code instructions for implementing the steps of a method according to at least one of claims 7 to 15 when executed by a processor.
18. A device for video encoding, comprising:
a processor configured to:
 determine whether a template-based coding mode is enabled for a current block;
 based on the determination of the template-based coding mode being enabled for the current block, identify a neighboring block based on a plurality of template sample values of the current block;
 obtain a value of a syntax element of the current block based on the identified neighboring block; and
 encode the current block based on the value of the syntax element.
19. The device of claim 18, wherein the processor is further configured to:
 register syntax element values of a plurality of neighboring blocks into a table; and
 obtain, based on the table, a syntax element value of the identified neighboring block that corresponds to the syntax element of the current block, wherein the value of the syntax element of the current block is obtained based on the syntax element value of the identified neighboring block.
20. The device of claim 18, wherein the processor is further configured to:

obtain a syntax element value of the identified neighboring block that corresponds to the syntax element of the current block; and

set the value of the syntax element of the current block to the syntax element value of the identified neighboring block.

21. The device of claim 18, wherein the processor is further configured to:

identify a plurality of neighboring blocks based on a plurality of template sample values of the plurality of neighboring blocks and the plurality of template sample values of the current block; and

obtain the value of the syntax element of the current block based on the identified plurality of neighboring blocks.

22. The device of claim 18, wherein the processor is further configured to:

compare a dimensionality of the neighboring block to a dimensionality of the current block;

based on the dimensionality of the neighboring block being larger than the current block, subsample a template of the neighboring block to equal a template size of the current block, wherein the neighboring block is identified based on a plurality of template sample values in the subsampled template of the neighboring block and the plurality of template sample values of the current block; and

based on the dimensionality of the current block being larger than the neighboring block, subsample a template of the current block to equal a template size of the neighboring block, wherein the neighboring block is identified based on a plurality of template sample values in the subsampled template of the current block.

23. The device of claim 18, wherein the template-based coding mode is enabled for a class of syntax elements for the current block, and wherein the processor is further configured to:

based on the determination of the template-based coding mode being enabled for the class of syntax elements for the current block, obtain values of the class of syntax elements of the current block based on corresponding syntax element values of the identified neighboring block; and

encode the current block using the values of the class of syntax elements of the current block.

24. A method for video encoding, comprising:

determining whether a template-based coding mode is enabled for a current block;

based on the determination of the template-based coding mode being enabled for the current block,
identifying a neighboring block based on a plurality of template sample values of the current block;
obtaining a value of a syntax element of the current block based on the identified neighboring block;
and
encoding the current block based on the value of the syntax element.

25. The method of claim 24, further comprising:

registering syntax element values of a plurality of neighboring blocks into a table; and
obtaining, based on the table, a syntax element value of the identified neighboring block that
corresponds to the syntax element of the current block, wherein the value of the syntax element of the current
block is obtained based on the syntax element value of the identified neighboring block.

26. The method of claim 24, further comprising:

obtaining a syntax element value of the identified neighboring block that corresponds to the syntax
element of the current block; and
setting the value of the syntax element of the current block to the syntax element value of the identified
neighboring block.

27. The method of claim 24, further comprising:

identifying a plurality of neighboring blocks based on a plurality of template sample values of the
plurality of neighboring blocks and the plurality of template sample values of the current block; and
obtaining the value of the syntax element of the current block based on the identified plurality of
neighboring blocks.

28. The method of claim 24, further comprising:

comparing a dimensionality of the neighboring block to a dimensionality of the current block;
based on the dimensionality of the neighboring block being larger than the current block, subsampling
a template of the neighboring block to equal a template size of the current block, wherein the neighboring block
is identified based on a plurality of template sample values in the subsampled template of the neighboring
block and the plurality of template sample values of the current block; and

based on the dimensionality of the current block being larger than the neighboring block, subsampling a template of the current block to equal a template size of the neighboring block, wherein the neighboring block is identified based on a plurality of template sample values in the subsampled template of the current block.

29. The method of claim 24, wherein the template-based coding mode is enabled for a class of syntax elements for the current block, further comprising:

based on the determination of the template-based coding mode being enabled for the class of syntax elements for the current block, obtaining values of the class of syntax elements of the current block based on corresponding syntax element values of the identified neighboring block; and

encoding the current block using the values of the class of syntax elements of the current block.

30. The device of claim 23 or the method of claim 29, wherein the class of syntax elements comprise intra prediction syntax elements.

31. The device of claim 23 or the method of claim 29, wherein the class of syntax elements comprise inter prediction syntax elements.

32. The device of claim 23 or the method of claim 29, wherein the class of syntax elements comprise transform syntax elements.

33. A device for video encoding, comprising:

a processor configured to:

determine whether to enable a template-based coding mode for a current block;

based on the determination to enable the template-based coding mode for the current block,

exclude signaling a syntax element for the current block; and

encode the current block based on the template-based coding mode.

34. The device of claim 33, wherein the template-based coding mode is enabled for a class of syntax elements for the current block, and wherein the processor is further configured to:

based on the determination to enable the template-based coding mode for the class of syntax elements for the current block, exclude signaling the class of syntax elements of the current block; and

encode the current block based on template-based coding mode.

35. A method for video encoding, comprising:

determining whether to enable template-based coding mode is for a current block;

based on the determination to enable of the template-based coding mode being for the current block, excluding signaling a syntax element for the current block; and

encoding the current block based on the template-based coding mode.

36. The method of claim 35, wherein the template-based coding mode is enabled for a class of syntax elements for the current block, further comprising:

based on the determination of the template-based coding mode being enabled for the class of syntax elements for the current block, excluding signaling the class of syntax elements of the current block; and

encoding the current block based on template-based coding mode.

37. The device of claim 33 or method of claim 35, wherein the determination of whether to enable the template-based coding mode for the current block is based on rate distortion optimization.

38. The device of claim 34 or method of claim 36, wherein the class of syntax elements comprise intra prediction syntax elements.

39. The device of claim 34 or method of claim 36, wherein the class of syntax elements comprise intra prediction syntax elements.

40. The device of claim 34 or method of claim 36, wherein the class of syntax elements comprise intra prediction syntax elements.

41. A computer program product which is stored on a non-transitory computer readable medium and comprises program code instructions for implementing the steps of a method according to at least one of claims 24 to 32 and 35 to 40 when executed by a processor.

42. A computer program comprising program code instructions for implementing the steps of a method according to at least one of claims 24 to 32 and 35 to 40 when executed by a processor.
43. A video data comprising information representative of the encoded output generated according to one of the methods of any of claims 24 to 32 and 35 to 40.