POSITION AND BLOCK CHARACTERISTICS DEPENDENT MATRIX INTRA PREDICTION FOR VIDEO CODING

[0001] This application claims priority to U.S. Patent Application No. 19/077,476, filed March 12, 2025, and U.S. Provisional Patent Application No. 63/632,421, filed April 10, 2024, the entire contents of each which are incorporated by reference herein. U.S. Patent Application No. 19/077,476, filed March 12, 2025, claims the benefit of U.S. Provisional Patent Application No. 63/632,421, filed April 10, 2024.

TECHNICAL FIELD

[0002] This disclosure relates to video encoding and video decoding.

BACKGROUND

[0003] Digital video capabilities can be incorporated into a wide range of devices, including digital televisions, digital direct broadcast systems, wireless broadcast systems, personal digital assistants (PDAs), laptop or desktop computers, tablet computers, e-book readers, digital cameras, digital recording devices, digital media players, video gaming devices, video game consoles, cellular or satellite radio telephones, so-called "smart phones," video teleconferencing devices, video streaming devices, and the like. Digital video devices implement video coding techniques, such as those described in the standards defined by MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Part 10, Advanced Video Coding (AVC), ITU-T H.265/High Efficiency Video Coding (HEVC), ITU-T H.266/Versatile Video Coding (VVC), and extensions of such standards, as well as proprietary video codecs/formats such as AOMedia Video 1 (AV1) that was developed by the Alliance for Open Media. The video devices may transmit, receive, encode, decode, and/or store digital video information more efficiently by implementing such video coding techniques.

[0004] Video coding techniques include spatial (intra-picture) prediction and/or temporal (inter-picture) prediction to reduce or remove redundancy inherent in video sequences. For block-based video coding, a video slice (e.g., a video picture or a portion of a video picture) may be partitioned into video blocks, which may also be referred to as coding tree units (CTUs), coding units (CUs) and/or coding nodes. Video blocks in an intra-coded (I) slice of a picture are encoded using spatial prediction with respect to reference samples in neighboring blocks in the same picture. Video blocks in an inter-coded (P or B) slice

of a picture may use spatial prediction with respect to reference samples in neighboring blocks in the same picture or temporal prediction with respect to reference samples in other reference pictures. Pictures may be referred to as frames, and reference pictures may be referred to as reference frames.

SUMMARY

[0005] In general, this disclosure describes techniques for intra prediction video coding. In particular, this disclosure describes techniques for prediction mode determination, including prediction mode determination for two or more matrix intra prediction (MIP). A video coder may determine a prediction mode from a set of possible prediction modes based on information from the current block and/or neighboring blocks to the current block. In one example, a video coder may determine a prediction mode for a block jointly using syntax elements and information from neighboring blocks. In another example, a video decoder may configured to perform decoder-side intra mode derivation (DIMD) using a whole prediction block, rather than a down-sampled prediction block, to determine a low frequency non-separable transform (LFSNT) transform set and/or a transpose flag. The techniques of this disclosure may improve coding efficiency by determining prediction modes using implicit information available at the decoder side.

[0006] In one example, this disclosure describes a method of coding video data, the method comprising coding a syntax element indicating a prediction mode type for a current block of video data, determining a prediction mode having the prediction mode type from a set of two more possible prediction modes based on one or more of information related to the current block or information related to neighboring blocks of the current block, and coding the current block using the prediction mode.

[0007] In another example, this disclosure describes an apparatus configured to code video data, the apparatus comprising a memory configured to store video data, and processing circuitry in communication with the memory, the processing circuitry configured to code a syntax element indicating a prediction mode type for a current block of video data, determine a prediction mode having the prediction mode type from a set of two more possible prediction modes based on one or more of information related to the current block or information related to neighboring blocks of the current block, and code the current block using the prediction mode.

[0008] In another example, this disclosure describes a non-transitory computer-readable storage medium storing instructions that, when executed cause one or more processors of a device configured to code video data to code a syntax element indicating a prediction mode type for a current block of video data, determine a prediction mode having the prediction mode type from a set of two more possible prediction modes based on one or more of information related to the current block or information related to neighboring blocks of the current block, and code the current block using the prediction mode.

[0009] The details of one or more examples are set forth in the accompanying drawings and the description below. Other features, objects, and advantages will be apparent from the description, drawings, and claims.

BRIEF DESCRIPTION OF DRAWINGS

[0010] FIG. 1 is a block diagram illustrating an example video encoding and decoding system that may perform the techniques of this disclosure.

[0011] FIG. 2 is a conceptual diagram illustrating example intra prediction modes that may be used in conjunction with the techniques of this disclosure.

[0012] FIG. 3 is a conceptual diagram illustrating one example of matrix intra prediction that may be used in conjunction with the techniques of this disclosure.

[0013] FIG. 4 is a conceptual diagram illustrating an example L-shaped template of samples around a current block that may be used in conjunction with the techniques of this disclosure.

[0014] FIG. 5 is a block diagram illustrating an example video encoder that may perform the techniques of this disclosure.

[0015] FIG. 6 is a block diagram illustrating an example video decoder that may perform the techniques of this disclosure.

[0016] FIG. 7 is a flowchart illustrating an example method for encoding a current block in accordance with the techniques of this disclosure.

[0017] FIG. 8 is a flowchart illustrating an example method for decoding a current block in accordance with the techniques of this disclosure.

[0018] FIG. 9 is a flowchart illustrating an example method for coding a current block in accordance with the techniques of this disclosure.

DETAILED DESCRIPTION

[0019] In some example video coding techniques, a video coder may code video data using a matrix intra prediction (MIP) mode. Current example MIP modes may not be suitable for all situations. As such, additional alternative MIP modes have been developed. However, simply adding signaling for additional coding modes may increase signaling overhead and decrease coding efficiency.

[0020] This disclosure describes techniques for determining a coding mode for a plurality of possible coding modes, including determining between two or more MIP coding modes. In general, this disclosure describes techniques for determining a prediction mode from among a plurality of prediction modes using neighboring and current block information, including the availability of neighboring samples and/or block size.

[0021] FIG. 1 is a block diagram illustrating an example video encoding and decoding system 100 that may perform the techniques of this disclosure. The techniques of this disclosure are generally directed to coding (encoding and/or decoding) video data. In general, video data includes any data for processing a video. Thus, video data may include raw, unencoded video, encoded video, decoded (e.g., reconstructed) video, and video metadata, such as signaling data.

[0022] As shown in FIG. 1, system 100 includes a source device 102 that provides encoded video data to be decoded and displayed by a destination device 116, in this example. In particular, source device 102 provides the video data to destination device 116 via a computer-readable medium 110. Source device 102 and destination device 116 may be or include any of a wide range of devices, such as desktop computers, notebook (i.e., laptop) computers, mobile devices, tablet computers, set-top boxes, telephone handsets such as smartphones, televisions, cameras, display devices, digital media players, video gaming consoles, video streaming device, broadcast receiver devices, or the like. In some cases, source device 102 and destination device 116 may be equipped for wireless communication, and thus may be referred to as wireless communication devices.

[0023] In the example of FIG. 1, source device 102 includes video source 104, memory 106, video encoder 200, and output interface 108. Destination device 116 includes input interface 122, video decoder 300, memory 120, and display device 118. In accordance with this disclosure, video encoder 200 of source device 102 and video decoder 300 of destination device 116 may be configured to apply the techniques for prediction mode

determination, including prediction mode determination for two or more matrix intra prediction modes. Thus, source device 102 represents an example of a video encoding device, while destination device 116 represents an example of a video decoding device. In other examples, a source device and a destination device may include other components or arrangements. For example, source device 102 may receive video data from an external video source, such as an external camera. Likewise, destination device 116 may interface with an external display device, rather than include an integrated display device.

[0024] System 100 as shown in FIG. 1 is merely one example. In general, any digital video encoding and/or decoding device may perform techniques for prediction mode determination. Source device 102 and destination device 116 are merely examples of such coding devices in which source device 102 generates coded video data for transmission to destination device 116. This disclosure refers to a "coding" device as a device that performs coding (encoding and/or decoding) of data. Thus, video encoder 200 and video decoder 300 represent examples of coding devices, in particular, a video encoder and a video decoder, respectively. In some examples, source device 102 and destination device 116 may operate in a substantially symmetrical manner such that each of source device 102 and destination device 116 includes video encoding and decoding components. Hence, system 100 may support one-way or two-way video transmission between source device 102 and destination device 116, e.g., for video streaming, video playback, video broadcasting, or video telephony.

[0025] In general, video source 104 represents a source of video data (i.e., raw, unencoded video data) and provides a sequential series of pictures (also referred to as "frames") of the video data to video encoder 200, which encodes data for the pictures. Video source 104 of source device 102 may include a video capture device, such as a video camera, a video archive containing previously captured raw video, and/or a video feed interface to receive video from a video content provider. As a further alternative, video source 104 may generate computer graphics-based data as the source video, or a combination of live video, archived video, and computer-generated video. In each case, video encoder 200 encodes the captured, pre-captured, or computer-generated video data. Video encoder 200 may rearrange the pictures from the received order (sometimes referred to as "display order") into a coding order for coding. Video encoder 200 may generate a bitstream including encoded video data. Source device 102 may then output the encoded video data via output interface 108 onto computer-readable medium 110 for reception and/or retrieval by, e.g., input interface 122 of destination device 116.

[0026] Memory 106 of source device 102 and memory 120 of destination device 116 represent general purpose memories. In some examples, memories 106, 120 may store raw video data, e.g., raw video from video source 104 and raw, decoded video data from video decoder 300. Additionally or alternatively, memories 106, 120 may store software instructions executable by, e.g., video encoder 200 and video decoder 300, respectively. Although memory 106 and memory 120 are shown separately from video encoder 200 and video decoder 300 in this example, it should be understood that video encoder 200 and video decoder 300 may also include internal memories for functionally similar or equivalent purposes. Furthermore, memories 106, 120 may store encoded video data, e.g., output from video encoder 200 and input to video decoder 300. In some examples, portions of memories 106, 120 may be allocated as one or more video buffers, e.g., to store raw, decoded, and/or encoded video data.

[0027] Computer-readable medium 110 may represent any type of medium or device capable of transporting the encoded video data from source device 102 to destination device 116. In one example, computer-readable medium 110 represents a communication medium to enable source device 102 to transmit encoded video data directly to destination device 116 in real-time, e.g., via a radio frequency network or computer-based network. Output interface 108 may modulate a transmission signal including the encoded video data, and input interface 122 may demodulate the received transmission signal, according to a communication standard, such as a wireless communication protocol. The communication medium may include any wireless or wired communication medium, such as a radio frequency (RF) spectrum or one or more physical transmission lines. The communication medium may form part of a packet-based network, such as a local area network, a wide-area network, or a global network such as the Internet. The communication medium may include routers, switches, base stations, or any other equipment that may be useful to facilitate communication from source device 102 to destination device 116.

[0028] In some examples, source device 102 may output encoded data from output interface 108 to storage device 112. Similarly, destination device 116 may access encoded data from storage device 112 via input interface 122. Storage device 112 may include any of a variety of distributed or locally accessed data storage media such as a hard drive, Blu-ray discs, DVDs, CD-ROMs, flash memory, volatile or non-volatile memory, or any other suitable digital storage media for storing encoded video data.

[0029] In some examples, source device 102 may output encoded video data to file server 114 or another intermediate storage device that may store the encoded video data generated by source device 102. Destination device 116 may access stored video data from file server 114 via streaming or download.

[0030] File server 114 may be any type of server device capable of storing encoded video data and transmitting that encoded video data to the destination device 116. File server 114 may represent a web server (e.g., for a website), a server configured to provide a file transfer protocol service (such as File Transfer Protocol (FTP) or File Delivery over Unidirectional Transport (FLUTE) protocol), a content delivery network (CDN) device, a hypertext transfer protocol (HTTP) server, a Multimedia Broadcast Multicast Service (MBMS) or Enhanced MBMS (eMBMS) server, and/or a network attached storage (NAS) device. File server 114 may, additionally or alternatively, implement one or more HTTP streaming protocols, such as Dynamic Adaptive Streaming over HTTP (DASH), HTTP Live Streaming (HLS), Real Time Streaming Protocol (RTSP), HTTP Dynamic Streaming, or the like.

[0031] Destination device 116 may access encoded video data from file server 114 through any standard data connection, including an Internet connection. This may include a wireless channel (e.g., a Wi-Fi connection), a wired connection (e.g., digital subscriber line (DSL), cable modem, etc.), or a combination of both that is suitable for accessing encoded video data stored on file server 114. Input interface 122 may be configured to operate according to any one or more of the various protocols discussed above for retrieving or receiving media data from file server 114, or other such protocols for retrieving media data.

[0032] Output interface 108 and input interface 122 may represent wireless transmitters/receivers, modems, wired networking components (e.g., Ethernet cards), wireless communication components that operate according to any of a variety of IEEE 802.11 standards, or other physical components. In examples where output interface 108 and input interface 122 include wireless components, output interface 108 and input interface 122 may be configured to transfer data, such as encoded video data, according to a cellular communication standard, such as 4G, 4G-LTE (Long-Term Evolution), LTE Advanced, 5G, or the like. In some examples where output interface 108 includes a wireless transmitter, output interface 108 and input interface 122 may be configured to transfer data, such as encoded video data, according to other wireless standards, such as an IEEE 802.11 specification, an IEEE 802.15 specification (e.g., ZigBee™), a

Bluetooth™ standard, or the like. In some examples, source device 102 and/or destination device 116 may include respective system-on-a-chip (SoC) devices. For example, source device 102 may include an SoC device to perform the functionality attributed to video encoder 200 and/or output interface 108, and destination device 116 may include an SoC device to perform the functionality attributed to video decoder 300 and/or input interface 122.

[0033] The techniques of this disclosure may be applied to video coding in support of any of a variety of multimedia applications, such as over-the-air television broadcasts, cable television transmissions, satellite television transmissions, Internet streaming video transmissions, such as dynamic adaptive streaming over HTTP (DASH), digital video that is encoded onto a data storage medium, decoding of digital video stored on a data storage medium, or other applications.

[0034] Input interface 122 of destination device 116 receives an encoded video bitstream from computer-readable medium 110 (e.g., a communication medium, storage device 112, file server 114, or the like). The encoded video bitstream may include signaling information defined by video encoder 200, which is also used by video decoder 300, such as syntax elements having values that describe characteristics and/or processing of video blocks or other coded units (e.g., slices, pictures, groups of pictures, sequences, or the like). Display device 118 displays decoded pictures of the decoded video data to a user. Display device 118 may represent any of a variety of display devices such as a liquid crystal display (LCD), a plasma display, an organic light emitting diode (OLED) display, or another type of display device.

[0035] Although not shown in FIG. 1, in some examples, video encoder 200 and video decoder 300 may each be integrated with an audio encoder and/or audio decoder (e.g., audio codec), and may include appropriate MUX-DEMUX units, or other hardware and/or software, to handle multiplexed streams including both audio and video in a common data stream. Example audio codecs may include AAC, AC-3, AC-4, ALAC, ALS, AMBE, AMR, AMR-WB (G.722.2), AMR-WB+, aptx (various versions), ATRAC, BroadVoice (BV16, BV32), CELT, Enhanced AC-3 (E-AC-3), EVS, FLAC, G.711, G.722, G.722.1, G.722.2 (AMR-WB). G.723.1, G.726, G.728, G.729, G.729.1, GSM-FR, HE-AAC, iLBC, iSAC, LA Lyra, Monkey's Audio, MP1, MP2 (MPEG-1, 2 Audio Layer II), MP3, Musepack, Nellymoser Asao, OptimFROG, Opus, Sac, Satin, SBC, SILK, Siren 7, Speex, SVOPC, True Audio (TTA), TwinVQ, USAC, Vorbis (Ogg), WavPack, and Windows Media Aud.

[0036] Video encoder 200 and video decoder 300 each may be implemented as any of a variety of suitable encoder and/or decoder circuitry that includes a processing system, such as one or more microprocessors, digital signal processors (DSPs), application specific integrated circuits (ASICs), field programmable gate arrays (FPGAs), discrete logic, software, hardware, firmware or any combinations thereof. When the techniques are implemented partially in software, a device may store instructions for the software in a suitable, non-transitory computer-readable medium and execute the instructions in hardware using one or more processors to perform the techniques of this disclosure. Each of video encoder 200 and video decoder 300 may be included in one or more encoders or decoders, either of which may be integrated as part of a combined encoder/decoder (CODEC) in a respective device. A device including video encoder 200 and/or video decoder 300 may implement video encoder 200 and/or video decoder 300 in processing circuitry such as an integrated circuit and/or a microprocessor. Such a device may be a wireless communication device, such as a cellular telephone, or any other type of device described herein.

[0037] Video encoder 200 and video decoder 300 may operate according to a video coding standard, such as ITU-T H.265, also referred to as High Efficiency Video Coding (HEVC) or extensions thereto, such as the multi-view and/or scalable video coding extensions. Alternatively, video encoder 200 and video decoder 300 may operate according to other proprietary or industry standards, such as ITU-T H.266, also referred to as Versatile Video Coding (VVC). In other examples, video encoder 200 and video decoder 300 may operate according to a proprietary video codec/format, such as AOMedia Video 1 (AV1), extensions of AV1, and/or successor versions of AV1 (e.g., AV2). In other examples, video encoder 200 and video decoder 300 may operate according to other proprietary formats or industry standards. The techniques of this disclosure, however, are not limited to any particular coding standard or format. In general, video encoder 200 and video decoder 300 may be configured to perform the techniques of this disclosure in conjunction with any video coding techniques that may implicitly determine a prediction mode from two or more prediction modes, including prediction mode determination for two or more matrix intra prediction modes.

[0038] In general, video encoder 200 and video decoder 300 may perform block-based coding of pictures. The term "block" generally refers to a structure including data to be processed (e.g., encoded, decoded, or otherwise used in the encoding and/or decoding process). For example, a block may include a two-dimensional matrix of samples of

luminance and/or chrominance data. In general, video encoder 200 and video decoder 300 may code video data represented in a YUV (e.g., Y, Cb, Cr) format. That is, rather than coding red, green, and blue (RGB) data for samples of a picture, video encoder 200 and video decoder 300 may code luminance and chrominance components, where the chrominance components may include both red hue and blue hue chrominance components. In some examples, video encoder 200 converts received RGB formatted data to a YUV representation prior to encoding, and video decoder 300 converts the YUV representation to the RGB format. Alternatively, pre- and post-processing units (not shown) may perform these conversions.

[0039] This disclosure may generally refer to coding (e.g., encoding and decoding) of pictures to include the process of encoding or decoding data of the picture. Similarly, this disclosure may refer to coding of blocks of a picture to include the process of encoding or decoding data for the blocks, e.g., prediction and/or residual coding. An encoded video bitstream generally includes a series of values for syntax elements representative of coding decisions (e.g., coding modes) and partitioning of pictures into blocks. Thus, references to coding a picture or a block should generally be understood as coding values for syntax elements forming the picture or block.

[0040] HEVC defines various blocks, including coding units (CUs), prediction units (PUs), and transform units (TUs). According to HEVC, a video coder (such as video encoder 200) partitions a coding tree unit (CTU) into CUs according to a quadtree structure. That is, the video coder partitions CTUs and CUs into four equal, non-overlapping squares, and each node of the quadtree has either zero or four child nodes. Nodes without child nodes may be referred to as "leaf nodes," and CUs of such leaf nodes may include one or more PUs and/or one or more TUs. The video coder may further partition PUs and TUs. For example, in HEVC, a residual quadtree (RQT) represents partitioning of TUs. In HEVC, PUs represent inter-prediction data, while TUs represent residual data. CUs that are intra-predicted include intra-prediction information, such as an intra-mode indication.

[0041] As another example, video encoder 200 and video decoder 300 may be configured to operate according to VVC. According to VVC, a video coder (such as video encoder 200) partitions a picture into a plurality of CTUs. Video encoder 200 may partition a CTU according to a tree structure, such as a quadtree-binary tree (QTBT) structure or Multi-Type Tree (MTT) structure. The QTBT structure removes the concepts of multiple partition types, such as the separation between CUs, PUs, and TUs of HEVC. A QTBT

structure includes two levels: a first level partitioned according to quadtree partitioning, and a second level partitioned according to binary tree partitioning. A root node of the QTBT structure corresponds to a CTU. Leaf nodes of the binary trees correspond to CUs.

[0042] In an MTT partitioning structure, blocks may be partitioned using a quadtree (QT) partition, a binary tree (BT) partition, and one or more types of triple tree (TT) (also called ternary tree (TT)) partitions. A triple or ternary tree partition is a partition where a block is split into three sub-blocks. In some examples, a triple or ternary tree partition divides a block into three sub-blocks without dividing the original block through the center. The partitioning types in MTT (e.g., QT, BT, and TT), may be symmetrical or asymmetrical.

[0043] When operating according to the AV1 codec, video encoder 200 and video decoder 300 may be configured to code video data in blocks. In AV1, the largest coding block that can be processed is called a superblock. In AV1, a superblock can be either 128x128 luma samples or 64x64 luma samples. However, in successor video coding formats (e.g., AV2), a superblock may be defined by different (e.g., larger) luma sample sizes. In some examples, a superblock is the top level of a block quadtree. Video encoder 200 may further partition a superblock into smaller coding blocks. Video encoder 200 may partition a superblock and other coding blocks into smaller blocks using square or non-square partitioning. Non-square blocks may include N/2xN, NxN/2, N/4xN, and NxN/4 blocks. Video encoder 200 and video decoder 300 may perform separate prediction and transform processes on each of the coding blocks.

[0044] AV1 also defines a tile of video data. A tile is a rectangular array of superblocks that may be coded independently of other tiles. That is, video encoder 200 and video decoder 300 may encode and decode, respectively, coding blocks within a tile without using video data from other tiles. However, video encoder 200 and video decoder 300 may perform filtering across tile boundaries. Tiles may be uniform or non-uniform in size. Tile-based coding may enable parallel processing and/or multi-threading for encoder and decoder implementations.

[0045] In some examples, video encoder 200 and video decoder 300 may use a single QTBT or MTT structure to represent each of the luminance and chrominance components, while in other examples, video encoder 200 and video decoder 300 may use two or more QTBT or MTT structures, such as one QTBT/MTT structure for the luminance component and another QTBT/MTT structure for both chrominance components (or two QTBT/MTT structures for respective chrominance components).

[0046] Video encoder 200 and video decoder 300 may be configured to use quadtree partitioning, QTBT partitioning, MTT partitioning, superblock partitioning, or other partitioning structures.

[0047] In some examples, a CTU includes a coding tree block (CTB) of luma samples, two corresponding CTBs of chroma samples of a picture that has three sample arrays, or a CTB of samples of a monochrome picture or a picture that is coded using three separate color planes and syntax structures used to code the samples. A CTB may be an NxN block of samples for some value of N such that the division of a component into CTBs is a partitioning. A component is an array or single sample from one of the three arrays (luma and two chroma) that compose a picture in 4:2:0, 4:2:2, or 4:4:4 color format or the array or a single sample of the array that compose a picture in monochrome format. In some examples, a coding block is an MxN block of samples for some values of M and N such that a division of a CTB into coding blocks is a partitioning.

[0048] The blocks (e.g., CTUs or CUs) may be grouped in various ways in a picture. As one example, a brick may refer to a rectangular region of CTU rows within a particular tile in a picture. A tile may be a rectangular region of CTUs within a particular tile column and a particular tile row in a picture. A tile column refers to a rectangular region of CTUs having a height equal to the height of the picture and a width specified by syntax elements (e.g., such as in a picture parameter set). A tile row refers to a rectangular region of CTUs having a height specified by syntax elements (e.g., such as in a picture parameter set) and a width equal to the width of the picture.

[0049] In some examples, a tile may be partitioned into multiple bricks, each of which may include one or more CTU rows within the tile. A tile that is not partitioned into multiple bricks may also be referred to as a brick. However, a brick that is a true subset of a tile may not be referred to as a tile. The bricks in a picture may also be arranged in a slice. A slice may be an integer number of bricks of a picture that may be exclusively contained in a single network abstraction layer (NAL) unit. In some examples, a slice includes either a number of complete tiles or only a consecutive sequence of complete bricks of one tile.

[0050] This disclosure may use "NxN" and "N by N" interchangeably to refer to the sample dimensions of a block (such as a CU or other video block) in terms of vertical and horizontal dimensions, e.g., 16x16 samples or 16 by 16 samples. In general, a 16x16 CU will have 16 samples in a vertical direction (y = 16) and 16 samples in a horizontal direction (x = 16). Likewise, an NxN CU generally has N samples in a vertical direction

and N samples in a horizontal direction, where N represents a nonnegative integer value. The samples in a CU may be arranged in rows and columns. Moreover, CUs need not necessarily have the same number of samples in the horizontal direction as in the vertical direction. For example, CUs may include NxM samples, where M is not necessarily equal to N.

[0051] Video encoder 200 encodes video data for CUs representing prediction and/or residual information, and other information. The prediction information indicates how the CU is to be predicted in order to form a prediction block for the CU. The residual information generally represents sample-by-sample differences between samples of the CU prior to encoding and the prediction block.

[0052] To predict a CU, video encoder 200 may generally form a prediction block for the CU through inter-prediction or intra-prediction. Inter-prediction generally refers to predicting the CU from data of a previously coded picture, whereas intra-prediction generally refers to predicting the CU from previously coded data of the same picture. To perform inter-prediction, video encoder 200 may generate the prediction block using one or more motion vectors. Video encoder 200 may generally perform a motion search to identify a reference block that closely matches the CU, e.g., in terms of differences between the CU and the reference block. Video encoder 200 may calculate a difference metric using a sum of absolute difference (SAD), sum of squared differences (SSD), mean absolute difference (MAD), mean squared differences (MSD), or other such difference calculations to determine whether a reference block closely matches the current CU. In some examples, video encoder 200 may predict the current CU using uni-directional prediction or bi-directional prediction.

[0053] Some examples of VVC also provide an affine motion compensation mode, which may be considered an inter-prediction mode. In affine motion compensation mode, video encoder 200 may determine two or more motion vectors that represent non-translational motion, such as zoom in or out, rotation, perspective motion, or other irregular motion types.

[0054] To perform intra-prediction, video encoder 200 may select an intra-prediction mode to generate the prediction block. Some examples of VVC provide sixty-seven intra-prediction modes, including various directional modes, as well as planar mode and DC mode. In general, video encoder 200 selects an intra-prediction mode that describes neighboring samples to a current block (e.g., a block of a CU) from which to predict samples of the current block. Such samples may generally be above, above and to the left,

or to the left of the current block in the same picture as the current block, assuming video encoder 200 codes CTUs and CUs in raster scan order (left to right, top to bottom).

[0055] Video encoder 200 encodes data representing the prediction mode for a current block. For example, for inter-prediction modes, video encoder 200 may encode data representing which of the various available inter-prediction modes is used, as well as motion information for the corresponding mode. For uni-directional or bi-directional inter-prediction, for example, video encoder 200 may encode motion vectors using advanced motion vector prediction (AMVP) or merge mode. Video encoder 200 may use similar modes to encode motion vectors for affine motion compensation mode.

[0056] AV1 includes two general techniques for encoding and decoding a coding block of video data. The two general techniques are intra prediction (e.g., intra frame prediction or spatial prediction) and inter prediction (e.g., inter frame prediction or temporal prediction). In the context of AV1, when predicting blocks of a current frame of video data using an intra prediction mode, video encoder 200 and video decoder 300 do not use video data from other frames of video data. For most intra prediction modes, video encoder 200 encodes blocks of a current frame based on the difference between sample values in the current block and predicted values generated from reference samples in the same frame. Video encoder 200 determines predicted values generated from the reference samples based on the intra prediction mode.

[0057] Following prediction, such as intra-prediction or inter-prediction of a block, video encoder 200 may calculate residual data for the block. The residual data, such as a residual block, represents sample by sample differences between the block and a prediction block for the block, formed using the corresponding prediction mode. Video encoder 200 may apply one or more transforms to the residual block, to produce transformed data in a transform domain instead of the sample domain. For example, video encoder 200 may apply a discrete cosine transform (DCT), an integer transform, a wavelet transform, or a conceptually similar transform to residual video data. Additionally, video encoder 200 may apply a secondary transform following the first transform, such as a mode-dependent non-separable secondary transform (MDNSST), a signal dependent transform, a Karhunen-Loeve transform (KLT), or the like. Video encoder 200 produces transform coefficients following application of the one or more transforms.

[0058] As noted above, following any transforms to produce transform coefficients, video encoder 200 may perform quantization of the transform coefficients. Quantization generally refers to a process in which transform coefficients are quantized to possibly

reduce the amount of data used to represent the transform coefficients, providing further compression. By performing the quantization process, video encoder 200 may reduce the bit depth associated with some or all of the transform coefficients. For example, video encoder 200 may round an n-bit value down to an m-bit value during quantization, where n is greater than m. In some examples, to perform quantization, video encoder 200 may perform a bitwise right-shift of the value to be quantized.

[0059] Following quantization, video encoder 200 may scan the transform coefficients, producing a one-dimensional vector from the two-dimensional matrix including the quantized transform coefficients. The scan may be designed to place higher energy (and therefore lower frequency) transform coefficients at the front of the vector and to place lower energy (and therefore higher frequency) transform coefficients at the back of the vector. In some examples, video encoder 200 may utilize a predefined scan order to scan the quantized transform coefficients to produce a serialized vector, and then entropy encode the quantized transform coefficients of the vector. In other examples, video encoder 200 may perform an adaptive scan. After scanning the quantized transform coefficients to form the one-dimensional vector, video encoder 200 may entropy encode the one-dimensional vector, e.g., according to context-adaptive binary arithmetic coding (CABAC). Video encoder 200 may also entropy encode values for syntax elements describing metadata associated with the encoded video data for use by video decoder 300 in decoding the video data.

[0060] To perform CABAC, video encoder 200 may assign a context within a context model to a symbol to be transmitted. The context may relate to, for example, whether neighboring values of the symbol are zero-valued or not. The probability determination may be based on a context assigned to the symbol.

[0061] Video encoder 200 may further generate syntax data, such as block-based syntax data, picture-based syntax data, and sequence-based syntax data, to video decoder 300, e.g., in a picture header, a block header, a slice header, or other syntax data, such as a sequence parameter set (SPS), picture parameter set (PPS), or video parameter set (VPS). Video decoder 300 may likewise decode such syntax data to determine how to decode corresponding video data.

[0062] In this manner, video encoder 200 may generate a bitstream including encoded video data, e.g., syntax elements describing partitioning of a picture into blocks (e.g., CUs) and prediction and/or residual information for the blocks. Ultimately, video decoder 300 may receive the bitstream and decode the encoded video data.

[0063] In general, video decoder 300 performs a reciprocal process to that performed by video encoder 200 to decode the encoded video data of the bitstream. For example, video decoder 300 may decode values for syntax elements of the bitstream using CABAC in a manner substantially similar to, albeit reciprocal to, the CABAC encoding process of video encoder 200. The syntax elements may define partitioning information for partitioning of a picture into CTUs, and partitioning of each CTU according to a corresponding partition structure, such as a QTBT structure, to define CUs of the CTU. The syntax elements may further define prediction and residual information for blocks (e.g., CUs) of video data.

[0064] The residual information may be represented by, for example, quantized transform coefficients. Video decoder 300 may inverse quantize and inverse transform the quantized transform coefficients of a block to reproduce a residual block for the block. Video decoder 300 uses a signaled prediction mode (intra- or inter-prediction) and related prediction information (e.g., motion information for inter-prediction) to form a prediction block for the block. Video decoder 300 may then combine the prediction block and the residual block (on a sample-by-sample basis) to reproduce the original block. Video decoder 300 may perform additional processing, such as performing a deblocking process to reduce visual artifacts along boundaries of the block.

[0065] This disclosure may generally refer to "signaling" certain information, such as syntax elements. The term "signaling" may generally refer to the communication of values for syntax elements and/or other data used to decode encoded video data. That is, video encoder 200 may signal values for syntax elements in the bitstream. In general, signaling refers to generating a value in the bitstream. As noted above, source device 102 may transport the bitstream to destination device 116 substantially in real time, or not in real time, such as might occur when storing syntax elements to storage device 112 for later retrieval by destination device 116.

[0066] Post VVC standardization, JVET been exploring technologies to further improve coding performance, and the exploration activity includes a corresponding test model reference software, called the Enhanced Compression Model (ECM). More detail on the ECM can be found in Coban, R.-L. Liao, K. Naser, J. Strom, L. Zhang, "Algorithm description of Enhanced Compression Model 10 (ECM 10)", 31st JVET Meeting, Geneva, CH, Jul. 2023, JVET-AE2025, and Y.-J. Chang et al. "Compression efficiency methods beyond VVC", 21st JVET Meeting, Geneva, CH, Jul. 2023, JVET-U0100.

[0067] In accordance with the techniques of this disclosure, as will be discussed in more detail below, video encoder 200 and video decoder 300 may be configured to receive a syntax element indicating a prediction mode type for a current block of video data, determine a prediction mode having the prediction mode type from a set of two more possible prediction modes based on one or more of information related to the current block or information related to neighboring blocks of the current block, and code the current block using the prediction mode.

[0068] **Regular Intra Prediction and Position Dependent Prediction Combination (PDPC)**

[0069] To capture the arbitrary edge directions presented in natural video, the number of directional intra modes in VVC/ECM was extended from 33 intra modes, as used in HEVC, to 65 intra modes. The new directional modes in VVC/ECM are depicted in FIG. 2. The planar and DC modes remain the same as in HEVC. These denser directional intra prediction modes apply for all block sizes and for both luma and chroma intra predictions.

[0070] In VVC and ECM, position-dependent prediction combination (PDPC) is an intra prediction technique applied to luma and chroma components. According to PDPC, a video coder (such as video encoder 200 or video decoder 300 of FIG. 1) may combine an original intra prediction signal with unfiltered or filtered boundary reference samples to generate a final prediction signal. According to VVC and ECM, the video coder may apply PDPC to the following intra modes without signaling: planar, DC, horizontal, vertical, and the angular modes with positive angle (modes having mode number less than 18 and greater than 50). In ECM, a gradient scheme of doing PDPC is extended from horizontal/vertical modes to some of the angular modes. Additional information on PDPC mode may be found in Y.-J. Chang et al. "Compression efficiency methods beyond VVC", 21st JVET Meeting, Geneva, CH, Jul. 2023, JVET-U0100, and B. Ray et al. "Unified PDPC for angular Intra modes", 17th JVET Meeting, Brussels, BE, Jan. 2020, JVET-Q0391.

[0071] **Matrix Intra Prediction (MIP)**

[0072] Matrix weighted intra prediction (MIP) coding mode is a newly added intra prediction technique into VVC. FIG. 3 is a conceptual diagram illustrating an example matrix intra prediction (MIP) process. According to MIP, to predict the samples of a rectangular block 140 of width W and height H, video encoder 200 or video decoder 300 may use neighboring reference samples 142 including one line of H reconstructed neighboring boundary samples left of the block and one line of W reconstructed

neighboring boundary samples above the block as input. If the reconstructed samples are unavailable, video encoder 200 or video decoder 300 may generate the reconstructed samples as is done in conventional intra prediction. The generation of the prediction signal may be based on the following three steps, which are averaging, matrix vector multiplication, and linear interpolation, as shown in FIG. 3.

[0073] Various different size values (e.g., three different size values) may be used for MIP. An index "idx" may be defined as follows. For idx in {0, 1, 2} there are 16, 12, and 6 matrices defined, which also defines the number of modes for that given idx. Additionally, each mode may be transposed, where the samples from left and above are swapped before performing matrix-vector multiplication. So, additionally, a transpose flag may be signaled (along with the mode signaling) when a coding unit (CU) is coded using MIP.

[0074] **Low Frequency Non-Separable Transform (LFNST) Mode Selection for MIP Coded Blocks**

[0075] A decoder-side intra mode derivation (DIMD) -based LFNST mode selection was adopted into ECM. One example is described in J.-Y. Huo, W.-H. Qiao, X. Hao, Y.-Z. Ma, F.-Z. Yang (Xidian Univ.), J. Ren, M. Li, L.-H. Xu (OPPO), "EE2-4.1: Modification of LFNST for MIP coded block", 28st JVET Meeting, Mainz, DE, Oct. 2022, JVET-AB0067. DIMD-based LFNST mode selection uses DIMD to derive an LFNST transform set and to determine an LFNST transpose flag. When a block uses MIP mode, the DIMD derivation process is performed on the down-sampled prediction block and the resulting mode is used to derived LFNST transform set and the transpose flag.

[0076] **Decoder-Side Intra Mode Derivation (DIMD)**

[0077] In intra prediction, besides the normal prediction mode, such as planar mode, DC mode, and one of a plurality of angular modes, video encoder 200 and video decoder 300 may use another mode called decoder-side intra mode derivation (DIMD). When operating according to DIMD, video decoder 300 may be configured to derive the coding mode from implicit information in the video bitstream. Video encoder 200 may perform the same process.

[0078] Video encoder 200 and video decoder 300 may derive a coding mode using a Histogram of Gradients (HoG). Here, the HoG is a vector of length 67 with each element denoting the magnitude of corresponding direction. The HoG creates an indication of a possible angular intra prediction mode. For a current coding unit (CU), video encoder 200 and video decoder 300 may compute the HoG with reconstructed samples from an above

reconstructed neighbor, a left reconstructed neighbor, and a top-left corner reconstructed neighbor.

[0079] Video encoder 200 and video decoder 300 may fuse the first two angular modes (e.g., mode 1 and mode2) from the HoG with the two highest magnitude with the planar mode as the final prediction mode derived from DIMD. The magnitude of mode1 and mode2 from DIMD may be referred to as mag1 and mag2. The weights of the fusion for mode1, mode2 and planar mode, respectively, are: $\frac{2mag1}{3(mag1+mag2)}$, $\frac{2mag2}{3(mag1+mag2)}$ and 1/3.

[0080] In VVC and the latest ECM, MIP modes do not make full use of available neighboring/current block information. This disclosure describes techniques to make use of more neighboring/current block information to benefit video signal compression tools, including MIP. In general, this disclosure describes techniques for determining a prediction mode from among a plurality of prediction modes using neighboring and current block information, including the availability of neighboring samples and/or block size.

[0081] **Example 1: Derive coding mode jointly from syntax elements and/or neighboring information**

[0082] In one example of the disclosure, video decoder 300 is configured to derive a prediction mode used to predict the current block jointly from one or more mode syntax elements that are signaled in a bit-stream and other information related to neighboring blocks and the current block. That is, video decoder 300 may determine the prediction mode from among a plurality of prediction modes based on both mode syntax elements and other information related to neighboring blocks and the current block. The information relating to the neighboring block and the current block may be information that is are already decoded from the bit-stream and/or derived in the decoding process. The techniques of this disclosure may also be performed by video encoder 200 in a reciprocal fashion, where information from the reconstruction process is used.

[0083] In one example, for a block with a certain amount of neighboring samples available (e.g., a number above some threshold), a prediction mode can be designed to make full use of the neighboring samples. While for those blocks without a certain amount of neighboring samples available (e.g., a number below some threshold), different prediction mode(s) can be designed to achieve good prediction results using the information that is available. In this context, available neighboring samples are samples that are reconstructed and available to be used for the reconstruction of the current block.

[0084] Each set of possible modes (e.g., $Mode_{allAvailable}$, $Mode_{availableCondition1}$, $Mode_{availableCondition2}$, … $Mode_{availableConditionN}$,) can be assigned to the same syntax element value(s) and video decoder 300 may determine the correct mode from set of possible modes by checking the availability of neighboring samples of the current block. In some examples, the set of possible modes may be related modes of the same general type. Examples below include a general MIP mode type (e.g., indicated by an MIP mode syntax element), where the set of possible MIP modes under the MIP mode type include the VVC/ECM MIP mode as well as at least one alternative MIP mode. The number of modes N can be 1, 2 or any positive number. While described with relationship to MIP mode, the techniques of this disclosure are applicable for any prediction mode type that may include two or more possible prediction modes in a set of possible modes that may be used.

[0085] Block sizes can also be used as a criterion for video decoder 300 to determine a coding mode from a set of possible coding modes. In this case, the set of modes that are assigned with the same syntax element value(s) include ($Mode_{size1}$, $Mode_{size2}$, … $Mode_{sizeN}$) and video decoder 300 may determine the mode within the set that is to be used by searching the set for the mode that is assigned to the current block size. The number N can be 2, 3 or any positive number. Please note that the 'sizeN' here does not have to be a single block size, but can be a set of multiple block sizes that are designed to use the same mode.

[0086] As one example, video decoder 300 may use the availability of the neighboring samples to differentiate between two modes in the mode derivation process of the current coding block. For a mode that is signaled/derived to be an MIP mode, in addition to the MIP mode used in VVC and ECM described above, an alternative MIP coding method may be available that is designed for blocks that have T1 neighboring reference lines available to the top boundary of the current block and have T2 neighboring reference lines available to the left boundary of the current block (e.g., including the T1*T2 corner block on the top-left size of the current block). This condition is referred to as 'neighbor condition' in the following text. Examples of the MIP alternative coding are described below. In this case, the mode used for each block that is signaled/derived to be MIP (e.g., by a syntax element indicating a general MIP type) is derived as follows:

- If the neighbor condition is met, the alternative MIP mode is used.
- Otherwise, because of not having the required number of neighboring samples available, the original MIP mode is used (e.g., the VVC/ECM

MIP mode). In some examples, not having the required number of neighboring samples could be due to being at the boundary of a picture, slice, tile, and/or virtual pipeline data unit (VPDU) or due to other constraints that make the neighboring reference lines unavailable.

[0087] As another example, the size of the current block may be used to differentiate between two modes in the mode derivation process of the current coding block. For a mode that is signaled/derived to be an MIP mode, in addition to the MIP mode used in VVC and ECM described above, an alternative MIP coding method may be available that is designed for blocks that are not considered to be 'narrow' (e.g., isNarrow = =False). For other blocks sizes, the original MIP mode is used (e.g. VVC/ECM MIP mode)

- As an example, the condition of a 'narrow' block is:

```
If( max(width, height) < 32 )
{
    isNarrow = False
}
Else if ( max(width, height)/min(width,height) >= 4)
{
    isNarrow = True
}
Else
{
    isNarrow = False
}
```

[0088] The examples mentioned above can be used together in any combination to create new embodiments of this disclosure.

[0089] As an example of the 'alternative MIP method' mentioned in the above examples, alternative matrices may be used to derive the prediction block from a causal template by performing matrix multiplication. The matrix multiplication takes M samples as input vector (M is the number of samples in the causal template) and produces N samples as output. N could be equal to or smaller than the target prediction block size W*H. In case of N < W*H, up-sampling is performed on the N output samples to generate the W*H prediction block.

[0090] FIG. 4 is a conceptual diagram illustrating an example L-shaped template of samples around a current block 130 that may be used in conjunction with the techniques

of this disclosure. As one example, for a block 130 with size W*H, the L-shaped neighbor reference samples include W_t * T1 top neighboring samples 132, H_t * T2 left neighboring samples134, and T1 * T2 corner samples 136. One example of the dimension values is W_t = 2* W, H_t = 2*H, T1 = T2 = 2. Another example is W_t = 2* W, H_t = 2*H, T1 = T2 = 1. In a video codec that uses the techniques of this disclosure, different numbers of reference lines in the L-shape templates can be used for different block sizes. As an example. W_t is always set to 2 * W, H_t is always set to 2*H, while T1=T2=1 is used for block sizes with max(W,H) >=32, and T1=T2=2 is used for blocks sizes with max(W,H) < 32.

[0091] The sizes of the L-shape template does not have to be the same as the sample availability check that used as 'neighbor condition' in this disclosure. As an example, the L-shape template for a 'neighbor condition' check may include W*T1 neighboring sample on top of the current block, the T2*H neighboring sample on the left side of the current block, together with the T1 x T2 corner neighboring samples. The L-shape template with W_t = 2 * W, H_t = 2 * H is used for matrix multiplication. If part of the L-shape template is not available, the nearest available sample in the same line is used to pad the missing area.

[0092] **Example 2**

[0093] In this example, video encoder 200 and video decoder 300 are configured to perform DIMD mode derivation using the whole prediction block, instead of a down-sampled prediction block utilized in ECM, in order to make full use of the current prediction block information to determine the LFNST transform set and transpose flag. Example techniques in ECM use a down-sampled block because of an MIP method use subsampling as a stage to derive the prediction.

[0094] Additionally, this DIMD derived mode can also be used for selecting the kernels for MTS (multiple separable transform set), and for most probable mode (MPM) list derivation for the future coding blocks (for which current coding block is a neighbor).

[0095] **Combination of Examples 1 and 2**

[0096] Examples 1 and 2 can be combined in a video codec. That is, video encoder 200 and video decoder 300 may be configured to perform techniques from both examples 1 and 2 described above.

[0097] The following describes a first example of the combination:

a. When the following neighbor condition and block size condition are both true, the alternative matrix multiplication-based coding method is used (e.g., the alternative MIP mode):
    i. If the W*T1 neighboring samples on the top side, H*T2 neighboring samples on the left side, and T1*T2 corner neighboring samples on the top-left side of the current blocks are available, where T1=T2=2, then the 'neighbor condition' is True.
    ii. If max(W,H) >= 32 and $\frac{\max(W,H)}{\min(W,H)} < 4$, the 'block size condition' is False. If max(W,H) > 32, the 'block size condition' is also False Otherwise the 'block size condition' is True.
    iii. The L-shaped causal template is configured to have $W_t = 2 * W$, $H_t = 2 * H$ and T1=T2=2 samples.
    iv. The output size N of the matrix multiplication is set to be W*H so that no up-sampling is needed for the prediction block.
b. When the alternative coding method is used, DIMD-based LFNST selection is performed on the full prediction block.

[0098] The following describes a second example of the combination:

a. The 'neighbor condition' and 'block size condition' is the same as the first example, except that for blocks with $\max(W,H) \geq 32$, neighbor condition and matrix setup is different:
    i. T1=T2=1 is used for both the neighbor condition and the L-shape template derivation.
    ii. Output of the matrix multiplication is set to $N = MIN(16,W) * MIN(16,H)$. A prediction block of W * H is obtained by up-sampling the N samples which corresponding to a $16 * 16$ block.
b. Same as the first example, DIMD derivation for LFNST is performed on the full prediction block when the alternative matrix-based prediction method is used.

[0099] In another example of the disclosure, whether or not to apply the alternative MIP mode, as proposed to the VVC/ECM MIP mode, is based on a maximum block size. That is for any block with width or height greater than 32, the alternative MIP mode is not applied. Instead, the VVC/ECM MIP mode is used.

[0100] In summary, in one example of the disclosure, video encoder 200 and video decoder 300 may be configured to code a syntax element indicating a prediction mode type for a current block of video data, determine a prediction mode having the prediction mode type from a set of two more possible prediction modes based on one or more of information related to the current block or information related to neighboring blocks of the current block, and code (e.g., encode or decode) the current block using the prediction mode. In one example, the prediction mode type is a matrix intra prediction (MIP) mode.

[0101] In one example of the disclosure, the information related to the current block is a block size. In another example of the disclosure, the information related to neighboring blocks of the current block is an availability of neighboring samples.

[0102] In one example, to determine the prediction mode, video encoder 200 and video decoder may be configured to determine the prediction mode to be a first prediction mode based on the availability of neighboring samples being above a threshold, and determine the prediction mode to be a second prediction mode based on the availability of neighboring samples being below or equal to a threshold.

[0103] In another example, to determine the prediction mode, video encoder 200 and video decoder may be configured to determine the prediction mode having the prediction mode type from the set of two more possible prediction modes based on the availability of neighboring samples and based on a size of the current block of video data.

[0104] In another example, video encoder 200 and video decoder may be configured to perform decoder side intra mode derivation process using a whole prediction block, wherein the whole prediction block is not down-sampled. In this example, video encoder 200 and video decoder may be further configured to determine one or more of a low frequency non-separate transform (LFNST) transform set or a transpose flag based on the decoder side intra mode derivation process.

[0105] FIG. 5 is a block diagram illustrating an example video encoder 200 that may perform the techniques of this disclosure. FIG. 5 is provided for purposes of explanation and should not be considered limiting of the techniques as broadly exemplified and described in this disclosure. For purposes of explanation, this disclosure describes video encoder 200 according to the techniques of VVC and HEVC. However, the techniques of this disclosure may be performed by video encoding devices that are configured to other video coding standards and video coding formats, such as AV1 and successors to the AV1 video coding format.

[0106] In the example of FIG. 5, video encoder 200 includes video data memory 230, mode selection unit 202, residual generation unit 204, transform processing unit 206, quantization unit 208, inverse quantization unit 210, inverse transform processing unit 212, reconstruction unit 214, filter unit 216, decoded picture buffer (DPB) 218, and entropy encoding unit 220. Any or all of video data memory 230, mode selection unit 202, residual generation unit 204, transform processing unit 206, quantization unit 208, inverse quantization unit 210, inverse transform processing unit 212, reconstruction unit 214, filter unit 216, DPB 218, and entropy encoding unit 220 may be implemented in one or more processors or in processing circuitry. For instance, the units of video encoder 200 may be implemented as one or more circuits or logic elements as part of hardware circuitry, or as part of a processor, ASIC, or FPGA. Moreover, video encoder 200 may include additional or alternative processors or processing circuitry to perform these and other functions.

[0107] Video data memory 230 is an example of a memory system that may store video data to be encoded by the components of video encoder 200. Video encoder 200 may receive the video data stored in video data memory 230 from, for example, video source 104 (FIG. 1). DPB 218 is an example of a memory system that may act as a reference picture memory that stores reference video data for use in prediction of subsequent video data by video encoder 200. Video data memory 230 and DPB 218 may each be formed by any of a variety of one or more memory devices or memory units, such as dynamic random access memory (DRAM), including synchronous DRAM (SDRAM), magnetoresistive RAM (MRAM), resistive RAM (RRAM), or other types of memory devices. Video data memory 230 and DPB 218 may be provided by the same memory device or separate memory devices. In various examples, video data memory 230 may be on-chip with other components of video encoder 200, as illustrated, or off-chip relative to those components.

[0108] In this disclosure, reference to video data memory 230 should not be interpreted as being limited to memory internal to video encoder 200, unless specifically described as such, or memory external to video encoder 200, unless specifically described as such. Rather, reference to video data memory 230 should be understood as reference memory that stores video data that video encoder 200 receives for encoding (e.g., video data for a current block that is to be encoded). Memory 106 of FIG. 1 may also provide temporary storage of outputs from the various units of video encoder 200.

[0109] The various units of FIG. 5 are illustrated to assist with understanding the operations performed by video encoder 200. The units may be implemented as fixed-function circuits, programmable circuits, or a combination thereof. Fixed-function circuits refer to circuits that provide particular functionality, and are preset on the operations that can be performed. Programmable circuits refer to circuits that can be programmed to perform various tasks, and provide flexible functionality in the operations that can be performed. For instance, programmable circuits may execute software or firmware that cause the programmable circuits to operate in the manner defined by instructions of the software or firmware. Fixed-function circuits may execute software instructions (e.g., to receive parameters or output parameters), but the types of operations that the fixed-function circuits perform are generally immutable. In some examples, one or more of the units may be distinct circuit blocks (fixed-function or programmable), and in some examples, one or more of the units may be integrated circuits.

[0110] Video encoder 200 may include arithmetic logic units (ALUs), elementary function units (EFUs), digital circuits, analog circuits, and/or programmable cores, formed from programmable circuits. In examples where the operations of video encoder 200 are performed using software executed by the programmable circuits, memory 106 (FIG. 1) may store the instructions (e.g., object code) of the software that video encoder 200 receives and executes, or another memory within video encoder 200 (not shown) may store such instructions.

[0111] Video data memory 230 is configured to store received video data. Video encoder 200 may retrieve a picture of the video data from video data memory 230 and provide the video data to residual generation unit 204 and mode selection unit 202. Video data in video data memory 230 may be raw video data that is to be encoded.

[0112] Mode selection unit 202 includes a motion estimation unit 222, a motion compensation unit 224, and an intra-prediction unit 226. Mode selection unit 202 may include additional functional units to perform video prediction in accordance with other prediction modes. As examples, mode selection unit 202 may include a palette unit, an intra-block copy unit (which may be part of motion estimation unit 222 and/or motion compensation unit 224), an affine unit, a linear model (LM) unit, or the like.

[0113] Mode selection unit 202 generally coordinates multiple encoding passes to test combinations of encoding parameters and resulting rate-distortion values for such combinations. The encoding parameters may include partitioning of CTUs into CUs, prediction modes for the CUs, transform types for residual data of the CUs, quantization

parameters for residual data of the CUs, and so on. Mode selection unit 202 may ultimately select the combination of encoding parameters having rate-distortion values that are better than the other tested combinations.

[0114] Video encoder 200 may partition a picture retrieved from video data memory 230 into a series of CTUs, and encapsulate one or more CTUs within a slice. Mode selection unit 202 may partition a CTU of the picture in accordance with a tree structure, such as the MTT structure, QTBT structure. superblock structure, or the quad-tree structure described above. As described above, video encoder 200 may form one or more CUs from partitioning a CTU according to the tree structure. Such a CU may also be referred to generally as a "video block" or "block."

[0115] In general, mode selection unit 202 also controls the components thereof (e.g., motion estimation unit 222, motion compensation unit 224, and intra-prediction unit 226) to generate a prediction block for a current block (e.g., a current CU, or in HEVC, the overlapping portion of a PU and a TU). For inter-prediction of a current block, motion estimation unit 222 may perform a motion search to identify one or more closely matching reference blocks in one or more reference pictures (e.g., one or more previously coded pictures stored in DPB 218). In particular, motion estimation unit 222 may calculate a value representative of how similar a potential reference block is to the current block, e.g., according to sum of absolute difference (SAD), sum of squared differences (SSD), mean absolute difference (MAD), mean squared differences (MSD), or the like. Motion estimation unit 222 may generally perform these calculations using sample-by-sample differences between the current block and the reference block being considered. Motion estimation unit 222 may identify a reference block having a lowest value resulting from these calculations, indicating a reference block that most closely matches the current block.

[0116] Motion estimation unit 222 may form one or more motion vectors (MVs) that defines the positions of the reference blocks in the reference pictures relative to the position of the current block in a current picture. Motion estimation unit 222 may then provide the motion vectors to motion compensation unit 224. For example, for uni-directional inter-prediction, motion estimation unit 222 may provide a single motion vector, whereas for bi-directional inter-prediction, motion estimation unit 222 may provide two motion vectors. Motion compensation unit 224 may then generate a prediction block using the motion vectors. For example, motion compensation unit 224 may retrieve data of the reference block using the motion vector. As another example, if

the motion vector has fractional sample precision, motion compensation unit 224 may interpolate values for the prediction block according to one or more interpolation filters. Moreover, for bi-directional inter-prediction, motion compensation unit 224 may retrieve data for two reference blocks identified by respective motion vectors and combine the retrieved data, e.g., through sample-by-sample averaging or weighted averaging.

[0117] When operating according to the AV1 video coding format, motion estimation unit 222 and motion compensation unit 224 may be configured to encode coding blocks of video data (e.g., both luma and chroma coding blocks) using translational motion compensation, affine motion compensation, overlapped block motion compensation (OBMC), and/or compound inter-intra prediction.

[0118] As another example, for intra-prediction, or intra-prediction coding, intra-prediction unit 226 may generate the prediction block from samples neighboring the current block. For example, for directional modes, intra-prediction unit 226 may generally mathematically combine values of neighboring samples and populate these calculated values in the defined direction across the current block to produce the prediction block. As another example, for DC mode, intra-prediction unit 226 may calculate an average of the neighboring samples to the current block and generate the prediction block to include this resulting average for each sample of the prediction block.

[0119] When operating according to the AV1 video coding format, intra-prediction unit 226 may be configured to encode coding blocks of video data (e.g., both luma and chroma coding blocks) using directional intra prediction, non-directional intra prediction, recursive filter intra prediction, chroma-from-luma (CFL) prediction, intra block copy (IBC), and/or color palette mode. Mode selection unit 202 may include additional functional units to perform video prediction in accordance with other prediction modes.

[0120] Mode selection unit 202 provides the prediction block to residual generation unit 204. Residual generation unit 204 receives a raw, unencoded version of the current block from video data memory 230 and the prediction block from mode selection unit 202. Residual generation unit 204 calculates sample-by-sample differences between the current block and the prediction block. The resulting sample-by-sample differences define a residual block for the current block. In some examples, residual generation unit 204 may also determine differences between sample values in the residual block to generate a residual block using residual differential pulse code modulation (RDPCM). In some examples, residual generation unit 204 may be formed using one or more subtractor circuits that perform binary subtraction.

[0121] In examples where mode selection unit 202 partitions CUs into PUs, each PU may be associated with a luma prediction unit and corresponding chroma prediction units. Video encoder 200 and video decoder 300 may support PUs having various sizes. As indicated above, the size of a CU may refer to the size of the luma coding block of the CU and the size of a PU may refer to the size of a luma prediction unit of the PU. Assuming that the size of a particular CU is 2Nx2N, video encoder 200 may support PU sizes of 2Nx2N or NxN for intra prediction, and symmetric PU sizes of 2Nx2N, 2NxN, Nx2N, NxN, or similar for inter prediction. Video encoder 200 and video decoder 300 may also support asymmetric partitioning for PU sizes of 2NxnU, 2NxnD, nLx2N, and nRx2N for inter prediction.

[0122] In examples where mode selection unit 202 does not further partition a CU into PUs, each CU may be associated with a luma coding block and corresponding chroma coding blocks. As above, the size of a CU may refer to the size of the luma coding block of the CU. The video encoder 200 and video decoder 300 may support CU sizes of 2Nx2N, 2NxN, or Nx2N.

[0123] For other video coding techniques such as an intra-block copy mode coding, an affine-mode coding, and linear model (LM) mode coding, as some examples, mode selection unit 202, via respective units associated with the coding techniques, generates a prediction block for the current block being encoded. In some examples, such as palette mode coding, mode selection unit 202 may not generate a prediction block, and instead generate syntax elements that indicate the manner in which to reconstruct the block based on a selected palette. In such modes, mode selection unit 202 may provide these syntax elements to entropy encoding unit 220 to be encoded.

[0124] As described above, residual generation unit 204 receives the video data for the current block and the corresponding prediction block. Residual generation unit 204 then generates a residual block for the current block. To generate the residual block, residual generation unit 204 calculates sample-by-sample differences between the prediction block and the current block.

[0125] Transform processing unit 206 applies one or more transforms to the residual block to generate a block of transform coefficients (referred to herein as a "transform coefficient block"). Transform processing unit 206 may apply various transforms to a residual block to form the transform coefficient block. For example, transform processing unit 206 may apply a discrete cosine transform (DCT), a directional transform, a Karhunen-Loeve transform (KLT), or a conceptually similar transform to a residual

block. In some examples, transform processing unit 206 may perform multiple transforms to a residual block, e.g., a primary transform and a secondary transform, such as a rotational transform. In some examples, transform processing unit 206 does not apply transforms to a residual block.

[0126] When operating according to AV1, transform processing unit 206 may apply one or more transforms to the residual block to generate a block of transform coefficients (referred to herein as a "transform coefficient block"). Transform processing unit 206 may apply various transforms to a residual block to form the transform coefficient block. For example, transform processing unit 206 may apply a horizontal/vertical transform combination that may include a discrete cosine transform (DCT), an asymmetric discrete sine transform (ADST), a flipped ADST (e.g., an ADST in reverse order), and an identity transform (IDTX). When using an identity transform, the transform is skipped in one of the vertical or horizontal directions. In some examples, transform processing may be skipped.

[0127] Quantization unit 208 may quantize the transform coefficients in a transform coefficient block, to produce a quantized transform coefficient block. Quantization unit 208 may quantize transform coefficients of a transform coefficient block according to a quantization parameter (QP) value associated with the current block. Video encoder 200 (e.g., via mode selection unit 202) may adjust the degree of quantization applied to the transform coefficient blocks associated with the current block by adjusting the QP value associated with the CU. Quantization may introduce loss of information, and thus, quantized transform coefficients may have lower precision than the original transform coefficients produced by transform processing unit 206.

[0128] Inverse quantization unit 210 and inverse transform processing unit 212 may apply inverse quantization and inverse transforms to a quantized transform coefficient block, respectively, to reconstruct a residual block from the transform coefficient block. Reconstruction unit 214 may produce a reconstructed block corresponding to the current block (albeit potentially with some degree of distortion) based on the reconstructed residual block and a prediction block generated by mode selection unit 202. For example, reconstruction unit 214 may add samples of the reconstructed residual block to corresponding samples from the prediction block generated by mode selection unit 202 to produce the reconstructed block.

[0129] Filter unit 216 may perform one or more filter operations on reconstructed blocks. For example, filter unit 216 may perform deblocking operations to reduce blockiness

artifacts along edges of CUs. Operations of filter unit 216 may be skipped, in some examples.

[0130] When operating according to AV1, filter unit 216 may perform one or more filter operations on reconstructed blocks. For example, filter unit 216 may perform deblocking operations to reduce blockiness artifacts along edges of CUs. In other examples, filter unit 216 may apply a constrained directional enhancement filter (CDEF), which may be applied after deblocking, and may include the application of non-separable, non-linear, low-pass directional filters based on estimated edge directions. Filter unit 216 may also include a loop restoration filter, which is applied after CDEF, and may include a separable symmetric normalized Wiener filter or a dual self-guided filter.

[0131] Video encoder 200 stores reconstructed blocks in DPB 218. For instance, in examples where operations of filter unit 216 are not performed, reconstruction unit 214 may store reconstructed blocks to DPB 218. In examples where operations of filter unit 216 are performed, filter unit 216 may store the filtered reconstructed blocks to DPB 218. Motion estimation unit 222 and motion compensation unit 224 may retrieve a reference picture from DPB 218, formed from the reconstructed (and potentially filtered) blocks, to inter-predict blocks of subsequently encoded pictures. In addition, intra-prediction unit 226 may use reconstructed blocks in DPB 218 of a current picture to intra-predict other blocks in the current picture.

[0132] In general, entropy encoding unit 220 may entropy encode syntax elements received from other functional components of video encoder 200. For example, entropy encoding unit 220 may entropy encode quantized transform coefficient blocks from quantization unit 208. As another example, entropy encoding unit 220 may entropy encode prediction syntax elements (e.g., motion information for inter-prediction or intra-mode information for intra-prediction) from mode selection unit 202. Entropy encoding unit 220 may perform one or more entropy encoding operations on the syntax elements, which are another example of video data, to generate entropy-encoded data. For example, entropy encoding unit 220 may perform a context-adaptive variable length coding (CAVLC) operation, a CABAC operation, a variable-to-variable (V2V) length coding operation, a syntax-based context-adaptive binary arithmetic coding (SBAC) operation, a Probability Interval Partitioning Entropy (PIPE) coding operation, an Exponential-Golomb encoding operation, or another type of entropy encoding operation on the data. In some examples, entropy encoding unit 220 may operate in bypass mode where syntax elements are not entropy encoded.

[0133] Video encoder 200 may output a bitstream that includes the entropy encoded syntax elements needed to reconstruct blocks of a slice or picture. In particular, entropy encoding unit 220 may output the bitstream.

[0134] In accordance with AV1, entropy encoding unit 220 may be configured as a symbol-to-symbol adaptive multi-symbol arithmetic coder. A syntax element in AV1 includes an alphabet of N elements, and a context (e.g., probability model) includes a set of N probabilities. Entropy encoding unit 220 may store the probabilities as n-bit (e.g., 15-bit) cumulative distribution functions (CDFs). Entropy encoding unit 220 may perform recursive scaling, with an update factor based on the alphabet size, to update the contexts.

[0135] The operations described above are described with respect to a block. Such description should be understood as being operations for a luma coding block and/or chroma coding blocks. As described above, in some examples, the luma coding block and chroma coding blocks are luma and chroma components of a CU. In some examples, the luma coding block and the chroma coding blocks are luma and chroma components of a PU.

[0136] In some examples, operations performed with respect to a luma coding block need not be repeated for the chroma coding blocks. As one example, operations to identify a motion vector (MV) and reference picture for a luma coding block need not be repeated for identifying a MV and reference picture for the chroma blocks. Rather, the MV for the luma coding block may be scaled to determine the MV for the chroma blocks, and the reference picture may be the same. As another example, the intra-prediction process may be the same for the luma coding block and the chroma coding blocks.

[0137] Video encoder 200 represents an example of a device configured to encode video data including a memory configured to store video data, and one or more processing units implemented in circuitry and configured to encode a syntax element indicating a prediction mode type for a current block of video data, determine a prediction mode having the prediction mode type from a set of two more possible prediction modes based on one or more of information related to the current block or information related to neighboring blocks of the current block, and encode the current block using the prediction mode.

[0138] FIG. 6 is a block diagram illustrating an example video decoder 300 that may perform the techniques of this disclosure. FIG. 6 is provided for purposes of explanation and is not limiting on the techniques as broadly exemplified and described in this

disclosure. For purposes of explanation, this disclosure describes video decoder 300 according to the techniques of VVC and HEVC. However, the techniques of this disclosure may be performed by video coding devices that are configured to other video coding standards.

[0139] In the example of FIG. 6, video decoder 300 includes coded picture buffer (CPB) memory 320, entropy decoding unit 302, prediction processing unit 304, inverse quantization unit 306, inverse transform processing unit 308, reconstruction unit 310, filter unit 312, and DPB 314. Any or all of CPB memory 320, entropy decoding unit 302, prediction processing unit 304, inverse quantization unit 306, inverse transform processing unit 308, reconstruction unit 310, filter unit 312, and DPB 314 may be implemented in one or more processors or in processing circuitry. For instance, the units of video decoder 300 may be implemented as one or more circuits or logic elements as part of hardware circuitry, or as part of a processor, ASIC, or FPGA. Moreover, video decoder 300 may include additional or alternative processors or processing circuitry to perform these and other functions.

[0140] Prediction processing unit 304 includes motion compensation unit 316 and intra-prediction unit 318. Prediction processing unit 304 may include additional units to perform prediction in accordance with other prediction modes. As examples, prediction processing unit 304 may include a palette unit, an intra-block copy unit (which may form part of motion compensation unit 316), an affine unit, a linear model (LM) unit, or the like. In other examples, video decoder 300 may include more, fewer, or different functional components.

[0141] When operating according to AV1, motion compensation unit 316 may be configured to decode coding blocks of video data (e.g., both luma and chroma coding blocks) using translational motion compensation, affine motion compensation, OBMC, and/or compound inter-intra prediction, as described above. Intra-prediction unit 318 may be configured to decode coding blocks of video data (e.g., both luma and chroma coding blocks) using directional intra prediction, non-directional intra prediction, recursive filter intra prediction, CFL, IBC, and/or color palette mode, as described above.

[0142] CPB memory 320 is an example of a memory system that may store video data, such as an encoded video bitstream, to be decoded by the components of video decoder 300. The video data stored in CPB memory 320 may be obtained, for example, from computer-readable medium 110 (FIG. 1). CPB memory 320 may include a CPB that stores encoded video data (e.g., syntax elements) from an encoded video bitstream. Also,

CPB memory 320 may store video data other than syntax elements of a coded picture, such as temporary data representing outputs from the various units of video decoder 300. DPB 314 is an example of a memory system that generally stores decoded pictures, which video decoder 300 may output and/or use as reference video data when decoding subsequent data or pictures of the encoded video bitstream. CPB memory 320 and DPB 314 may each be formed by any of a variety of memory devices or memory units, such as DRAM, including SDRAM, MRAM, RRAM, or other types of memory devices. CPB memory 320 and DPB 314 may be provided by the same memory device or separate memory devices. In various examples, CPB memory 320 may be on-chip with other components of video decoder 300, or off-chip relative to those components.

[0143] Additionally or alternatively, in some examples, video decoder 300 may retrieve coded video data from memory 120 (FIG. 1). That is, memory 120 may store data as discussed above with CPB memory 320. Likewise, memory 120 may store instructions to be executed by video decoder 300, when some or all of the functionality of video decoder 300 is implemented in software to be executed by processing circuitry of video decoder 300.

[0144] The various units shown in FIG. 6 are illustrated to assist with understanding the operations performed by video decoder 300. The units may be implemented as fixed-function circuits, programmable circuits, or a combination thereof. Similar to FIG. 5, fixed-function circuits refer to circuits that provide particular functionality, and are preset on the operations that can be performed. Programmable circuits refer to circuits that can be programmed to perform various tasks, and provide flexible functionality in the operations that can be performed. For instance, programmable circuits may execute software or firmware that cause the programmable circuits to operate in the manner defined by instructions of the software or firmware. Fixed-function circuits may execute software instructions (e.g., to receive parameters or output parameters), but the types of operations that the fixed-function circuits perform are generally immutable. In some examples, one or more of the units may be distinct circuit blocks (fixed-function or programmable), and in some examples, one or more of the units may be integrated circuits.

[0145] Video decoder 300 may include ALUs, EFUs, digital circuits, analog circuits, and/or programmable cores formed from programmable circuits. In examples where the operations of video decoder 300 are performed by software executing on the

programmable circuits, on-chip or off-chip memory may store instructions (e.g., object code) of the software that video decoder 300 receives and executes.

[0146] Entropy decoding unit 302 may receive encoded video data from the CPB and entropy decode the video data to reproduce syntax elements. Prediction processing unit 304, inverse quantization unit 306, inverse transform processing unit 308, reconstruction unit 310, and filter unit 312 may generate decoded video data based on the syntax elements extracted from the bitstream.

[0147] In general, video decoder 300 reconstructs a picture on a block-by-block basis. Video decoder 300 may perform a reconstruction operation on each block individually (where the block currently being reconstructed, i.e., decoded, may be referred to as a "current block").

[0148] Entropy decoding unit 302 may entropy decode syntax elements defining quantized transform coefficients of a quantized transform coefficient block, as well as transform information, such as a quantization parameter (QP) and/or transform mode indication(s). Inverse quantization unit 306 may use the QP associated with the quantized transform coefficient block to determine a degree of quantization and, likewise, a degree of inverse quantization for inverse quantization unit 306 to apply. Inverse quantization unit 306 may, for example, perform a bitwise left-shift operation to inverse quantize the quantized transform coefficients. Inverse quantization unit 306 may thereby form a transform coefficient block including transform coefficients.

[0149] After inverse quantization unit 306 forms the transform coefficient block, inverse transform processing unit 308 may apply one or more inverse transforms to the transform coefficient block to generate a residual block associated with the current block. For example, inverse transform processing unit 308 may apply an inverse DCT, an inverse integer transform, an inverse Karhunen-Loeve transform (KLT), an inverse rotational transform, an inverse directional transform, or another inverse transform to the transform coefficient block.

[0150] Furthermore, prediction processing unit 304 generates a prediction block according to prediction information syntax elements that were entropy decoded by entropy decoding unit 302. For example, if the prediction information syntax elements indicate that the current block is inter-predicted, motion compensation unit 316 may generate the prediction block. In this case, the prediction information syntax elements may indicate a reference picture in DPB 314 from which to retrieve a reference block, as well as a motion vector identifying a location of the reference block in the reference

picture relative to the location of the current block in the current picture. Motion compensation unit 316 may generally perform the inter-prediction process in a manner that is substantially similar to that described with respect to motion compensation unit 224 (FIG. 5).

[0151] As another example, if the prediction information syntax elements indicate that the current block is intra-predicted, intra-prediction unit 318 may generate the prediction block according to an intra-prediction mode indicated by the prediction information syntax elements. Again, intra-prediction unit 318 may generally perform the intra-prediction process in a manner that is substantially similar to that described with respect to intra-prediction unit 226 (FIG. 5). Intra-prediction unit 318 may retrieve data of neighboring samples to the current block from DPB 314.

[0152] Reconstruction unit 310 may reconstruct the current block using the prediction block and the residual block. For example, reconstruction unit 310 may add samples of the residual block to corresponding samples of the prediction block to reconstruct the current block.

[0153] Filter unit 312 may perform one or more filter operations on reconstructed blocks. For example, filter unit 312 may perform deblocking operations to reduce blockiness artifacts along edges of the reconstructed blocks. Operations of filter unit 312 are not necessarily performed in all examples.

[0154] Video decoder 300 may store the reconstructed blocks in DPB 314. For instance, in examples where operations of filter unit 312 are not performed, reconstruction unit 310 may store reconstructed blocks to DPB 314. In examples where operations of filter unit 312 are performed, filter unit 312 may store the filtered reconstructed blocks to DPB 314. As discussed above, DPB 314 may provide reference information, such as samples of a current picture for intra-prediction and previously decoded pictures for subsequent motion compensation, to prediction processing unit 304. Moreover, video decoder 300 may output decoded pictures (e.g., decoded video) from DPB 314 for subsequent presentation on a display device, such as display device 118 of FIG. 1.

[0155] In this manner, video decoder 300 represents an example of a video decoding device including a memory configured to store video data, and one or more processing units implemented in circuitry and configured decode a syntax element indicating a prediction mode type for a current block of video data, determine a prediction mode having the prediction mode type from a set of two more possible prediction modes based on one or more of information related to the current block or information related to

neighboring blocks of the current block, and decode the current block using the prediction mode.

[0156] FIG. 7 is a flowchart illustrating an example method for encoding a current block in accordance with the techniques of this disclosure. The current block may be or include a current CU. Although described with respect to video encoder 200 (FIGS. 1 and 5), it should be understood that other devices may be configured to perform a method similar to that of FIG. 7.

[0157] In this example, video encoder 200 initially predicts the current block (400). For example, video encoder 200 may form a prediction block for the current block. Video encoder 200 may then calculate a residual block for the current block (402). To calculate the residual block, video encoder 200 may calculate a difference between the original, unencoded block and the prediction block for the current block. Video encoder 200 may then transform the residual block and quantize transform coefficients of the residual block (404). Next, video encoder 200 may scan the quantized transform coefficients of the residual block (406). During the scan, or following the scan, video encoder 200 may entropy encode the transform coefficients (408). For example, video encoder 200 may encode the transform coefficients using CAVLC or CABAC. Video encoder 200 may then output the entropy encoded data of the block (410).

[0158] FIG. 8 is a flowchart illustrating an example method for decoding a current block of video data in accordance with the techniques of this disclosure. The current block may be or include a current CU. Although described with respect to video decoder 300 (FIGS. 1 and 6), it should be understood that other devices may be configured to perform a method similar to that of FIG. 8.

[0159] Video decoder 300 may receive entropy encoded data for the current block, such as entropy encoded prediction information and entropy encoded data for transform coefficients of a residual block corresponding to the current block (500). Video decoder 300 may entropy decode the entropy encoded data to determine prediction information for the current block and to reproduce transform coefficients of the residual block (502). Video decoder 300 may predict the current block (504), e.g., using an intra- or inter-prediction mode as indicated by the prediction information for the current block, to calculate a prediction block for the current block. Video decoder 300 may then inverse scan the reproduced transform coefficients (506), to create a block of quantized transform coefficients. Video decoder 300 may then inverse quantize the transform coefficients and apply an inverse transform to the transform coefficients to produce a residual block (508).

Video decoder 300 may ultimately decode the current block by combining the prediction block and the residual block (510).

[0160] FIG. 9 is a flowchart illustrating an example method for coding a current block in accordance with the techniques of this disclosure. The techniques of FIG. 9 may be performed by one or more components of video encoder 200 and/or video decoder 300.

[0161] In one example, video encoder 200 and/or video decoder 300 may be configured to code (e.g., encode or decode, respectively) a syntax element indicating a prediction mode type for a current block of video data (900), determine a prediction mode having the prediction mode type from a set of two more possible prediction modes based on one or more of information related to the current block or information related to neighboring blocks of the current block (910), and code the current block using the prediction mode (920). In one example, the prediction mode type is a matrix intra prediction (MIP) mode. The information related to the current block may be a block size. In another example, in addition to or instead of the block size, the information related to neighboring blocks of the current block is an availability of neighboring samples.

[0162] In another example, to determine the prediction mode, video encoder 200 and/or video decoder 300 may be further configured to determine the prediction mode to be a first prediction mode based on the availability of neighboring samples being above a threshold, and determine the prediction mode to be a second prediction mode based on the availability of neighboring samples being below or equal to a threshold. To determine the prediction mode, video encoder 200 and/or video decoder 300 may be further configured to determine the prediction mode having the prediction mode type from the set of two more possible prediction modes based on the availability of neighboring samples and based on a size of the current block of video data.

[0163] In another example, video encoder 200 and/or video decoder 300 may be further configured to perform decoder side intra mode derivation process using a whole prediction block, wherein the whole prediction block is not down-sampled.

[0164] In another example, video encoder 200 and/or video decoder 300 may be further configured to determine one or more of a low frequency non-separate transform (LFNST) transform set or a transpose flag based on the decoder side intra mode derivation process.

[0165] The following numbered clauses illustrate one or more aspects of the devices and techniques described in this disclosure.

[0166] Aspect 1A. A method of coding video data, the method comprising: receiving one or more syntax elements for a current block; deriving a prediction mode for the current

block jointly based on the one or more syntax element and other information related to the current block or neighboring blocks of the current block; and coding the block using the prediction mode.

[0167] Aspect 2A. The method of Aspect 1A, wherein the current block is coded using matrix intra prediction.

[0168] Aspect 3A. A method of coding video data, the method comprising: receiving a current block of video data; and performing decoder side intra mode derivation using a whole prediction block.

[0169] Aspect 4A. The method of Aspect 3A, wherein the whole prediction block is not down-sampled.

[0170] Aspect 5A. The method of Aspect 3A, wherein the decoder side intra mode derivation determines a low frequency non-separate transform (LFNST) transform set and a transpose flag.

[0171] Aspect 6A. A method of any combination of Aspects 1A-5A.

[0172] Aspect 7A. The method of any of Aspects 1A–6A, wherein coding comprises decoding.

[0173] Aspect 8A. The method of any of Aspects 1A–6A, wherein coding comprises encoding.

[0174] Aspect 9A. A device for coding video data, the device comprising one or more means for performing the method of any of Aspects 1A–8A.

[0175] Aspect 10A. The device of Aspect 9A, wherein the one or more means comprise one or more processors implemented in circuitry.

[0176] Aspect 11A. The device of any of Aspects 9A and 10A, further comprising a memory to store the video data.

[0177] Aspect 12A. The device of any of Aspects 9A-11A, further comprising a display configured to display decoded video data.

[0178] Aspect 13A. The device of any of Aspects 9A-12A, wherein the device comprises one or more of a camera, a computer, a mobile device, a broadcast receiver device, or a set-top box.

[0179] Aspect 14A. The device of any of Aspects 9A-13A, wherein the device comprises a video decoder.

[0180] Aspect 15A. The device of any of Aspects 9A-14A, wherein the device comprises a video encoder.

[0181] Aspect 16A. A computer-readable storage medium having stored thereon instructions that, when executed, cause one or more processors to perform the method of any of Aspects 1A–8A.

[0182] Aspect 1B. A method of coding video data, the method comprising: coding a syntax element indicating a prediction mode type for a current block of video data; determining a prediction mode having the prediction mode type from a set of two more possible prediction modes based on one or more of information related to the current block or information related to neighboring blocks of the current block; and coding the current block using the prediction mode.

[0183] Aspect 2B. The method of Aspect 1B, wherein the prediction mode type is a matrix intra prediction (MIP) mode.

[0184] Aspect 3B. The method of any of Aspects 1B-2B, wherein the information related to the current block is a block size.

[0185] Aspect 4B. The method of any of Aspects 1B-3B, wherein the information related to neighboring blocks of the current block is an availability of neighboring samples.

[0186] Aspect 5B. Thet method of Aspect 4B, wherein determining the prediction mode comprises: determining the prediction mode to be a first prediction mode based on the availability of neighboring samples being above a threshold; and determining the prediction mode to be a second prediction mode based on the availability of neighboring samples being below or equal to a threshold.

[0187] Aspect 6B. The method of Aspect 5B, wherein determining the prediction mode comprises: determining the prediction mode having the prediction mode type from the set of two more possible prediction modes based on the availability of neighboring samples and based on a size of the current block of video data.

[0188] Aspect 7B. The method of any of Aspects 1B-6B, further comprising: performing decoder side intra mode derivation process using a whole prediction block, wherein the whole prediction block is not down-sampled.

[0189] Aspect 8B. The method of Aspect 7B, further comprising: determining one or more of a low frequency non-separate transform (LFNST) transform set or a transpose flag based on the decoder side intra mode derivation process.

[0190] Aspect 9B. The method of any of Aspects 1B-8B, wherein coding the current block using the prediction mode comprises encoding the current block using the prediction mode.

[0191] Aspect 10B. The method of any of Aspects 1B-8B, wherein coding the current block using the prediction mode comprises decoding the current block using the prediction mode.

[0192] Aspect 11B. An apparatus configured to code video data, the apparatus comprising: a memory configured to store video data; and processing circuitry in communication with the memory, the processing circuitry configured to: code a syntax element indicating a prediction mode type for a current block of video data; determine a prediction mode having the prediction mode type from a set of two more possible prediction modes based on one or more of information related to the current block or information related to neighboring blocks of the current block; and code the current block using the prediction mode.

[0193] Aspect 12B. The apparatus of Aspect 11B, wherein the prediction mode type is a matrix intra prediction (MIP) mode.

[0194] Aspect 13B. The apparatus of any of Aspects 11B-12B, wherein the information related to the current block is a block size.

[0195] Aspect 14B. The apparatus of any of Aspects 11B-13B, wherein the information related to neighboring blocks of the current block is an availability of neighboring samples.

[0196] Aspect 15B. Thet apparatus of Aspect 14B, wherein to determine the prediction mode, the processing circuitry is further configured to: determine the prediction mode to be a first prediction mode based on the availability of neighboring samples being above a threshold; and determine the prediction mode to be a second prediction mode based on the availability of neighboring samples being below or equal to a threshold.

[0197] Aspect 16B. The apparatus of Aspect 15B, wherein to determine the prediction mode, the processing circuitry is further configured to: determine the prediction mode having the prediction mode type from the set of two more possible prediction modes based on the availability of neighboring samples and based on a size of the current block of video data.

[0198] Aspect 17B. The apparatus of any of Aspects 11B-16B, wherein the processing circuitry is further configured to: perform decoder side intra mode derivation process using a whole prediction block, wherein the whole prediction block is not down-sampled.

[0199] Aspect 18B. The apparatus of Aspect 17B, wherein the processing circuitry is further configured to: determine one or more of a low frequency non-separate transform

(LFNST) transform set or a transpose flag based on the decoder side intra mode derivation process.

[0200] Aspect 19B. A non-transitory computer-readable storage medium storing instructions that, when executed cause one or more processors of a device configured to code video data to: code a syntax element indicating a prediction mode type for a current block of video data; determine a prediction mode having the prediction mode type from a set of two more possible prediction modes based on one or more of information related to the current block or information related to neighboring blocks of the current block; and code the current block using the prediction mode.

[0201] Aspect 20B. The non-transitory computer-readable storage medium of Aspect 19B, wherein the prediction mode type is a matrix intra prediction (MIP) mode.

[0202] It is to be recognized that depending on the example, certain acts or events of any of the techniques described herein can be performed in a different sequence, may be added, merged, or left out altogether (e.g., not all described acts or events are necessary for the practice of the techniques). Moreover, in certain examples, acts or events may be performed concurrently, e.g., through multi-threaded processing, interrupt processing, or multiple processors, rather than sequentially.

[0203] In one or more examples, the functions described may be implemented in hardware, software, firmware, or any combination thereof. If implemented in software, the functions may be stored on or transmitted over as one or more instructions or code on a computer-readable medium and executed by a hardware-based processing unit. Computer-readable media may include computer-readable storage media, which corresponds to a tangible medium such as data storage media, or communication media including any medium that facilitates transfer of a computer program from one place to another, e.g., according to a communication protocol. In this manner, computer-readable media generally may correspond to (1) tangible computer-readable storage media which is non-transitory or (2) a communication medium such as a signal or carrier wave. Data storage media may be any available media that can be accessed by one or more computers or one or more processors to retrieve instructions, code and/or data structures for implementation of the techniques described in this disclosure. A computer program product may include a computer-readable medium.

[0204] By way of example, and not limitation, such computer-readable storage media may include one or more of RAM, ROM, EEPROM, CD-ROM or other optical disk storage, magnetic disk storage, or other magnetic storage devices, flash memory, or any

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

[0205] Instructions may be executed by one or more processors, such as one or more DSPs, general purpose microprocessors, ASICs, FPGAs, or other equivalent integrated or discrete logic circuitry. Accordingly, the terms "processor" and "processing circuitry," as used herein may refer to any of the foregoing structures or any other structure suitable for implementation of the techniques described herein. In addition, in some aspects, the functionality described herein may be provided within dedicated hardware and/or software modules configured for encoding and decoding, or incorporated in a combined codec. Also, the techniques could be fully implemented in one or more circuits or logic elements.

[0206] The techniques of this disclosure may be implemented in a wide variety of devices or apparatuses, including a wireless handset, an integrated circuit (IC) or a set of ICs (e.g., a chip set). Various components, modules, or units are described in this disclosure to emphasize functional aspects of devices configured to perform the disclosed techniques, but do not necessarily require realization by different hardware units. Rather, as described above, various units may be combined in a codec hardware unit or provided by a collection of interoperative hardware units, including one or more processors as described above, in conjunction with suitable software and/or firmware.

[0207] Various examples have been described. These and other examples are within the scope of the following claims.

WHAT IS CLAIMED IS:

1. A method of coding video data, the method comprising:

coding a syntax element indicating a prediction mode type for a current block of video data;

determining a prediction mode having the prediction mode type from a set of two more possible prediction modes based on one or more of information related to the current block or information related to neighboring blocks of the current block; and

coding the current block using the prediction mode.

2. The method of claim 1, wherein the prediction mode type is a matrix intra prediction (MIP) mode.

3. The method of claim 1, wherein the information related to the current block is a block size.

4. The method of claim 1, wherein the information related to neighboring blocks of the current block is an availability of neighboring samples.

5. Thet method of claim 4, wherein determining the prediction mode comprises:

determining the prediction mode to be a first prediction mode based on the availability of neighboring samples being above a threshold; and

determining the prediction mode to be a second prediction mode based on the availability of neighboring samples being below or equal to a threshold.

6. The method of claim 5, wherein determining the prediction mode comprises:

determining the prediction mode having the prediction mode type from the set of two more possible prediction modes based on the availability of neighboring samples and based on a size of the current block of video data.

7. The method of claim 1, further comprising:

performing decoder side intra mode derivation process using a whole prediction block, wherein the whole prediction block is not down-sampled.

8. The method of claim 7, further comprising:

determining one or more of a low frequency non-separate transform (LFNST) transform set or a transpose flag based on the decoder side intra mode derivation process.

9. The method of claim 1, wherein coding the current block using the prediction mode comprises encoding the current block using the prediction mode.

10. The method of claim 1, wherein coding the current block using the prediction mode comprises decoding the current block using the prediction mode.

11. An apparatus configured to code video data, the apparatus comprising:

a memory configured to store video data; and

processing circuitry in communication with the memory, the processing circuitry configured to:

code a syntax element indicating a prediction mode type for a current block of video data;

determine a prediction mode having the prediction mode type from a set of two more possible prediction modes based on one or more of information related to the current block or information related to neighboring blocks of the current block; and

code the current block using the prediction mode.

12. The apparatus of claim 11, wherein the prediction mode type is a matrix intra prediction (MIP) mode.

13. The apparatus of claim 11, wherein the information related to the current block is a block size.

14. The apparatus of claim 11, wherein the information related to neighboring blocks of the current block is an availability of neighboring samples.

15. Thet apparatus of claim 14, wherein to determine the prediction mode, the processing circuitry is further configured to:

determine the prediction mode to be a first prediction mode based on the availability of neighboring samples being above a threshold; and

determine the prediction mode to be a second prediction mode based on the availability of neighboring samples being below or equal to a threshold.

16. The apparatus of claim 15, wherein to determine the prediction mode, the processing circuitry is further configured to:

determine the prediction mode having the prediction mode type from the set of two more possible prediction modes based on the availability of neighboring samples and based on a size of the current block of video data.

17. The apparatus of claim 11, wherein the processing circuitry is further configured to:

perform decoder side intra mode derivation process using a whole prediction block, wherein the whole prediction block is not down-sampled.

18. The apparatus of claim 17, wherein the processing circuitry is further configured to:

determine one or more of a low frequency non-separate transform (LFNST) transform set or a transpose flag based on the decoder side intra mode derivation process.

19. A non-transitory computer-readable storage medium storing instructions that, when executed cause one or more processors of a device configured to code video data to:

code a syntax element indicating a prediction mode type for a current block of video data;

determine a prediction mode having the prediction mode type from a set of two more possible prediction modes based on one or more of information related to the current block or information related to neighboring blocks of the current block; and

code the current block using the prediction mode.

20. The non-transitory computer-readable storage medium of claim 19, wherein the prediction mode type is a matrix intra prediction (MIP) mode.