

BI-PREDICTIVE MERGE MODE BASED ON UNI-PREDICTIVE NEIGHBORS IN VIDEO CODING

[0001] This application claims the benefit of U.S. Provisional Application Number 61/454,862, filed on March 21, 2011 and U.S. Provisional Application Number 61/502,703, filed on June 29, 2011, the entire content each of which is incorporated herein by reference.

TECHNICAL FIELD

[0002] This disclosure relates to video coding techniques used to compress video data and, more particularly, video coding modes used in video compression.

BACKGROUND

[0003] Digital video capabilities can be incorporated into a wide range of video devices, including digital televisions, digital direct broadcast systems, wireless communication devices such as wireless telephone handsets, wireless broadcast systems, personal digital assistants (PDAs), laptop or desktop computers, tablet computers, digital cameras, digital recording devices, video gaming devices, video game consoles, personal multimedia players, and the like. Such video devices may implement video compression techniques, such as those described in MPEG-2, MPEG-4, or ITU-T H.264/MPEG-4, Part 10, Advanced Video Coding (AVC), in order to compress video data. Video compression techniques perform spatial and/or temporal prediction to reduce or remove redundancy inherent in video sequences. New video standards, such as the High Efficiency Video Coding (HEVC) standard being developed by the “Joint Collaborative Team – Video Coding” (JCTVC), which is a collaboration between MPEG and ITU-T, continue to emerge and evolve. The emerging HEVC standard is sometimes referred to as H.265.

[0004] These and other video coding standards and techniques use block-based video coding. Block-based video coding techniques divide the video data of a video frame (or portion thereof) into video blocks and then encode the video blocks using predictive block-based compression techniques. The video blocks may be further divided into video block partitions. The video blocks (or partitions thereof) may be referred to as coding units (CUs) and may be encoded using one or more video-specific encoding

techniques as well as general data compression techniques. Different modes may be selected and used to code the video blocks.

[0005] With the emerging HEVC standard, largest coding units (LCUs) may be divided into smaller and smaller CUs according to a quadtree partitioning scheme. The CUs may be predicted based on so-called prediction units (PUs), which can have partition sizes corresponding to the size of the CUs or smaller than the size of the CUs, such that multiple PUs can be used to predict a given CU.

[0006] Different modes may be used to encode the CUs. For example, different intra coding modes may be used to code the CUs based on predictive data within the same frame or slice so as to exploit spatial redundancy within a video frame. Alternatively, inter coding modes may be used to code CUs based on predictive data from another frame or slice, so as to exploit temporal redundancy across frames of a video sequence. After the predictive coding is performed according to a selected mode, transform coding may then be performed, such as discrete cosine transforms (DCT), integer transforms or the like. With HEVC, the transform coding may occur with respect to transform units (TUs), which can also have varying transform sizes in the HEVC standard.

Quantization of the transform coefficients, scanning of the quantized transform coefficients, and entropy coding may also be performed. Syntax information is signaled with encoded video data, e.g., in a video slice header or video block header, in order to inform the decoder how to decode the video data. Among other things, the syntax information may identify the mode that was used in the video coding of different video blocks.

[0007] Merge mode is a specific inter coding mode used in video compression. With merge mode, the motion vector of a neighboring video block is inherited for a current video block being coded. In some cases, merge mode causes a current video block to inherit the motion vector of a pre-defined neighbor, and in other cases, an index value may be used to identify the specific neighbor from which the current video block inherits its motion vector (e.g., top, top right, left, left bottom or co-located from a temporally adjacent frame).

SUMMARY

[0008] This disclosure describes a bi-predictive merge mode in which a video block coded in the bi-predictive merge mode inherits its motion information from two

different neighboring blocks, wherein the two different neighboring blocks were each encoded in a uni-predictive mode. Bi-predictive coding may improve the ability to achieve compression or improve the video quality at a given level of compression. However, in some cases, there may be no (or few) neighbors that were encoded in a bi-predictive mode, thus rendering bi-prediction unavailable (or limited) with respect to merge mode encoding. The described bi-predictive merge mode may increase the number of bi-predictive candidates that can be used in the context of merge mode coding by allowing two separate uni-predicted neighbors to be used to define bi-predictive motion information for a video block.

[0009] In one example, this disclosure describes a method of decoding video data. The method comprises receiving one or more syntax elements for a current video block, wherein the current video block is encoded according to a bi-predictive merge mode, and based on the one or more syntax elements, identifying two different neighboring video blocks coded in uni-predictive modes. The method also comprises using motion information of the two different neighboring video blocks to decode a current video block according to the bi-predictive merge mode.

[0010] In another example, this disclosure describes a method of encoding video data. The method comprises selecting a bi-predictive merge mode for encoding a current video block, identifying two different neighboring video blocks coded in uni-predictive modes, using motion information of the two different neighboring video blocks to encode the current video block according to the bi-predictive merge mode, and generating one or more syntax elements to identify the two different neighboring video blocks to a video decoder.

[0011] In another example, this disclosure describes a video decoding device that decodes video data. The video decoding device comprises a video decoder configured to receive one or more syntax elements for a current video block, wherein the current video block is encoded according to a bi-predictive merge mode, and based on the one or more syntax elements, identify two different neighboring video blocks coded in uni-predictive modes. The video decoder is configured to use motion information of the two different neighboring video blocks to decode a current video block according to the bi-predictive merge mode.

[0012] In another example, this disclosure describes a video encoding device comprising a video encoder configured to select a bi-predictive merge mode for encoding a current video block, identify two different neighboring video blocks coded

in uni-predictive modes, use motion information of the two different neighboring video blocks to encode the current video block according to the bi-predictive merge mode, and generate one or more syntax elements to identify the two different neighboring video blocks to a video decoder.

[0013] In another example, this disclosure describes a device for decoding video data, the device comprising means for receiving one or more syntax elements for a current video block, wherein the current video block is encoded according to a bi-predictive merge mode, means for identifying two different neighboring video blocks coded in uni-predictive modes based on the one or more syntax elements, and means for using motion information of the two different neighboring video blocks to decode a current video block according to the bi-predictive merge mode.

[0014] In another example, this disclosure describes a device for encoding video data, the device comprising means for selecting a bi-predictive merge mode for encoding a current video block, means for identifying two different neighboring video blocks coded in uni-predictive modes, means for using motion information of the two different neighboring video blocks to encode the current video block according to the bi-predictive merge mode, and means for generating one or more syntax elements to identify the two different neighboring video blocks to a video decoder.

[0015] The techniques described in this disclosure may be implemented in hardware, software, firmware, or any combination thereof. For example, various techniques may be implemented or executed by one or more processors. As used herein, a processor may refer to a microprocessor, an application specific integrated circuit (ASIC), a field programmable gate array (FPGA), a digital signal processor (DSP), or other equivalent integrated or discrete logic circuitry. Software may be executed by one or more processors. Software comprising instructions to execute the techniques may be initially stored in a computer-readable medium and loaded and executed by a processor.

[0016] Accordingly, this disclosure also contemplates computer-readable storage media comprising instructions to cause a processor to perform any the techniques described in this disclosure. In some cases, the computer-readable storage medium may form part of a computer program storage product, which may be sold to manufacturers and/or used in a device. The computer program product may include the computer-readable medium, and in some cases, may also include packaging materials.

[0017] In particular, this disclosure also describes a computer-readable medium comprising instructions that upon execution cause a processor to decode video data,

wherein the instructions cause the processor to upon receiving one or more syntax elements for a current video block, wherein the current video block is encoded according to a bi-predictive merge mode, identify two different neighboring video blocks coded in uni-predictive modes based on the one or more syntax elements, use motion information of the two different neighboring video blocks to decode a current video block according to the bi-predictive merge mode.

[0018] In yet another example, this disclosure describes a computer-readable medium comprising instructions that upon execution cause a processor to encode video data, wherein the instructions cause the processor to select a bi-predictive merge mode for encoding a current video block, identify two different neighboring video blocks coded in uni-predictive modes, use motion information of the two different neighboring video blocks to encode the current video block according to the bi-predictive merge mode, and generate one or more syntax elements to identify the two different neighboring video blocks to a video decoder.

[0019] The details of one or more aspects of the disclosure are set forth in the accompanying drawings and the description below. Other features, objects, and advantages of the techniques described in this disclosure will be apparent from the description and drawings, and from the claims.

BRIEF DESCRIPTION OF DRAWINGS

[0020] FIG. 1 is a block diagram illustrating one exemplary video encoding and decoding system that may implement one or more of the techniques of this disclosure.

[0021] FIG. 2 is a conceptual diagram illustrating quadtree partitioning of coding units (CUs) consistent with the techniques of this disclosure.

[0022] FIG. 3 is a conceptual diagram illustrating some possible relationships between CUs, prediction units (PUs) and transform units (TUs) consistent with the techniques of this disclosure.

[0023] FIG. 4 is block diagram illustrating a video encoder that may implement techniques of this disclosure.

[0024] FIG. 5 is a block diagram illustrating an exemplary prediction unit of an encoder, consistent with one or more examples of this disclosure

[0025] FIG. 6 is a block diagram illustrating a video decoder that may implement techniques of this disclosure.

[0026] FIG. 7 is a conceptual diagram illustrating the location of different neighboring video blocks relative to current video block such that the current video block may use information of one or more different neighboring video blocks in a bi-predictive merge mode, consistent with this disclosure.

[0027] FIGS. 8 and 9 are flow diagrams illustrating techniques consistent with this disclosure.

DETAILED DESCRIPTION

[0028] In most video coding systems, motion estimation and motion compensation are used to reduce the temporal redundancy in a video sequence, in order to achieve data compression. In this case, a motion vector can be generated so as to identify a predictive block of video data, e.g., from another video frame or slice, which can be used to predict the values of the current video block being coded. The values of the predictive video block are subtracted from the values of the current video block to produce a block of residual data. The motion vector is communicated from the encoder to the decoder, along with the residual data. The decoder can locate the same predictive block (based on the motion vector) and reconstruct the encoded video block by combining the residual data with the data of the predictive block. Many other compression techniques can also be used, such as transforms and entropy coding, to further improve the video compression.

[0029] The motion estimation process is usually done at the encoder. Motion information (such as motion vectors, motion vector indexes, prediction directions, or other information) can be encoded and transmitted from the encoder to the decoder so that the decoder can identify the same predictive block that was used to encode a given video block. Many different encoding modes can be used to allow for different types of temporal prediction between two different frames or different types of spatial prediction within a given frame.

[0030] With so-called merge mode, the motion information of a neighboring video block is inherited for a current video block being coded. In this case, the motion vector itself is not transmitted for a video block coded in merge mode. Rather, an index value may be used to identify the neighbor from which the current video block inherits its motion vector (and possibly other motion information). For example, motion information may be inherited from a top neighbor, a top right neighbor, a left neighbor,

a left bottom neighbor or a co-located temporal neighbor from a temporally adjacent frame.

[0031] With most merge modes, if the neighbor is encoded in a uni-predictive mode, then the current video block inherits one motion vector. If the neighbor is encoded in a bi-predictive mode, then the current video block inherits two motion vectors. In such examples, a block being coded in merge mode is limited by the motion information of its neighbors. Uni-prediction and bi-prediction are sometimes referred to as uni-directional (P) prediction and bi-directional (B) prediction, but the term “directional” is generally misplaced because with modern video coding standards, bi-prediction is simply based on two different lists of predictive data and direction is not mandated. In other words, the data in the two different lists for bi-prediction may come from previous or subsequent frames, and does not need to be bi-directional from both previous and subsequent frames respectively. For this reason, this disclosure uses the terms uni-prediction and bi-prediction rather than the terms uni-directional prediction and bi-directional prediction.

[0032] Bi-predictive coding may improve the ability to achieve compression or improve the video quality at a given level of compression. However, in some cases, there may be no (or few) neighbors that were encoded in a bi-predictive mode, thus rendering bi-prediction unavailable (or limited) in merge mode encoding. For example, with conventional merge mode, if there is no bi-predictive mode in any of the neighboring blocks, the current block may lose the opportunity to exploit the benefits that may arise from bi-prediction.

[0033] This disclosure describes a bi-predictive merge mode as an extension or addition to merge mode techniques. More specifically, this disclosure describes a bi-predictive merge mode that inherits motion information from two different neighboring blocks, wherein the two different neighboring blocks were each encoded in a uni-predictive mode. The described bi-predictive merge mode may increase the number of bi-predictive candidates that can be used in the context of merge mode coding.

[0034] FIG. 1 is a block diagram illustrating an exemplary video encoding and decoding system 10 that may implement techniques of this disclosure. As shown in FIG. 1, system 10 includes a source device 12 that transmits encoded video to a destination device 16 via a communication channel 15. Source device 12 and destination device 16 may comprise any of a wide range of devices. In some cases, source device 12 and destination device 16 may comprise wireless communication

device handsets, such as so-called cellular or satellite radiotelephones. The techniques of this disclosure, however, which apply generally to the encoding and decoding of video blocks in bi-predictive merge mode may be applied to non-wireless devices including video encoding and/or decoding capabilities. Source device 12 and destination device 16 are merely examples of coding devices that can support the techniques described herein.

[0035] In the example of FIG. 1, source device 12 may include a video source 20, a video encoder 22, a modulator/demodulator (modem) 23 and a transmitter 24. Destination device 16 may include a receiver 26, a modem 27, a video decoder 28, and a display device 30. In accordance with this disclosure, video encoder 22 of source device 12 may be configured to encode one or more video blocks according to a bi-predictive merge mode. With bi-predictive merge mode, a video block inherits its motion information from two different neighboring blocks, wherein the two different neighboring blocks were each encoded in a uni-predictive mode. Syntax elements may be generated at video encoder 22 in order to identify the two different neighboring video blocks coded in uni-predictive modes. In this way, a video decoder can reconstruct a bi-predictive video block based on motion information of the two different neighboring video blocks identified by the syntax elements.

[0036] More specifically, video encoder 22 may select a bi-predictive merge mode for encoding a current video block, and identify two different neighboring video blocks coded in uni-predictive modes. Video encoder 22 may use motion information of the two different neighboring video blocks to encode the current video block according to the bi-predictive merge mode, and generate one or more syntax elements to identify the two different neighboring video blocks to a video decoder.

[0037] Video source 20 may comprise a video capture device, such as a video camera, a video archive containing previously captured video, a video feed from a video content provider or another source of video. As a further alternative, video source 20 may generate computer graphics-based data as the source video, or a combination of live video, archived video, and computer-generated video. In some cases, if video source 20 is a video camera, source device 12 and destination device 16 may form so-called camera phones or video phones. In each case, the captured, pre-captured or computer-generated video may be encoded by video encoder 22.

[0038] Once the video data is encoded by video encoder 22, the encoded video information may then be modulated by modem 23 according to a communication

standard, e.g., such as code division multiple access (CDMA), orthogonal frequency division multiplexing (OFDM) or any other communication standard or technique. The encoded and modulated data can then be transmitted to destination device 16 via transmitter 24. Modem 23 may include various mixers, filters, amplifiers or other components designed for signal modulation. Transmitter 24 may include circuits designed for transmitting data, including amplifiers, filters, and one or more antennas. Receiver 26 of destination device 16 receives information over channel 15, and modem 27 demodulates the information.

[0039] The video decoding process performed by video decoder 28 may include reciprocal techniques to the encoding techniques performed by video encoder 22. In particular, video decoder 28 may receive one or more syntax elements for a current video block, wherein the current video block is encoded according to a bi-predictive merge mode, and based on the one or more syntax elements, identify two different neighboring video blocks coded in uni-predictive modes. Video decoder may use motion information of the two different neighboring video blocks to decode a current video block according to the bi-predictive merge mode.

[0040] Communication channel 15 may comprise any wireless or wired communication medium, such as a radio frequency (RF) spectrum or one or more physical transmission lines, or any combination of wireless and wired media. Communication channel 15 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. Communication channel 15 generally represents any suitable communication medium, or collection of different communication media, for transmitting video data from source device 12 to destination device 16. Again, FIG. 1 is merely exemplary and the techniques of this disclosure may apply to video coding settings (e.g., video encoding or video decoding) that do not necessarily include any data communication between the encoding and decoding devices. In other examples, data could be retrieved from a local memory, streamed over a network, or the like.

[0041] In some cases, video encoder 22 and video decoder 28 may operate substantially according to a video compression standard such as the emerging HEVC standard. However, the techniques of this disclosure may also be applied in the context of a variety of other video coding standards, including some old standards, or new or emerging standards. Although not shown in FIG. 1, in some cases, video encoder 22 and video decoder 28 may each be integrated with an audio encoder and decoder, and

may include appropriate MUX-DEMUX units, or other hardware and software, to handle encoding of both audio and video in a common data stream or separate data streams. If applicable, MUX-DEMUX units may conform to the ITU H.223 multiplexer protocol, or other protocols such as the user datagram protocol (UDP).

[0042] Video encoder 22 and video decoder 28 each may be implemented 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 combinations thereof. Each of video encoder 22 and video decoder 28 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 mobile device, subscriber device, broadcast device, server, or the like. In this disclosure, the term coder refers to an encoder, a decoder, or CODEC, and the terms coder, encoder, decoder and CODEC all refer to specific machines designed for the coding (encoding and/or decoding) of video data consistent with this disclosure.

[0043] In some cases, devices 12, 16 may operate in a substantially symmetrical manner. For example, each of devices 12, 16 may include video encoding and decoding components. Hence, system 10 may support one-way or two-way video transmission between video devices 12, 16, e.g., for video streaming, video playback, video broadcasting, or video telephony.

[0044] During the encoding process, video encoder 22 may execute a number of coding techniques or operations. In general, video encoder 22 operates on blocks of video data consistent with the HEVC standard. Consistent with HEVC, the video blocks are referred to as coding units (CUs) and many CUs exist within individual video frames (or other independently defined units of video, such as slices). Frames, slices, portions of frames, groups of pictures, or other data structures may be defined as units of video information that include a plurality of CUs. The CUs may have varying sizes consistent with the HEVC standard, and the bitstream may define largest coding units (LCUs) as the largest size of CU. The bi-predictive merge mode may be used to encode LCUs, CUs, or possibly other types of video blocks. With the HEVC standard, LCUs may be divided into smaller and smaller CUs according to a quadtree partitioning scheme, and the different CUs that are defined in the scheme may be further partitioned into so-called prediction units (PUs). The LCUs, CUs, and PUs are all video blocks within the meaning of this disclosure.

[0045] Video encoder 22 may perform predictive coding in which a video block being coded (e.g., a PU of a CU within an LCU) is compared to one or more predictive candidates in order to identify a predictive block. This process of predictive coding may be intra (in which case the predictive data is generated based on neighboring intra data within the same video frame or slice) or inter (in which case the predictive data is generated based on video data in previous or subsequent frames or slices). Many different coding modes may be supported, and video encoder 22 may select a desirable video coding mode. According to this disclosure, at least some video blocks may be coded using the bi-predictive merge mode described herein.

[0046] After generating the predictive block, the differences between the current video block being coded and the predictive block are coded as a residual block, and prediction syntax (such as a motion vector in the case of inter coding, or a predictive mode in the case of intra coding) is used to identify the predictive block. Moreover, with the bi-predictive merge mode described herein, the prediction syntax (e.g., syntax elements) may identify two different neighboring video blocks to a video decoder. Accordingly, the decoder can identify two different neighboring video blocks coded in uni-predictive modes, based on the syntax elements, and use motion information of the two different neighboring video blocks to decode a current video block according to the bi-predictive merge mode.

[0047] The residual block may be transformed and quantized. Transform techniques may comprise a DCT process or conceptually similar process, integer transforms, wavelet transforms, or other types of transforms. In a DCT process, as an example, the transform process converts a set of pixel values (e.g., residual pixel values) into transform coefficients, which may represent the energy of the pixel values in the frequency domain. The HEVC standard allows for transformations according to transformation units (TUs), which may be different for different CUs. The TUs are typically sized based on the size of PUs within a given CU defined for a partitioned LCU, although this may not always be the case. The TUs are typically the same size or smaller than the PUs.

[0048] Quantization may be applied to the transform coefficients, and generally involves a process that limits the number of bits associated with any given transform coefficient. More specifically, quantization may be applied according to a quantization parameter (QP) defined at the LCU level. Accordingly, the same level of quantization may be applied to all transform coefficients in the TUs associated with different PUs of

CUs within an LCU. However, rather than signal the QP itself, a change (i.e., a delta) in the QP may be signaled with the LCU to indicate the change in QP relative to that of a previous LCU.

[0049] Following transform and quantization, entropy coding may be performed on the quantized and transformed residual video blocks. Syntax elements may also be included in the entropy coded bitstream. In general, entropy coding comprises one or more processes that collectively compress a sequence of quantized transform coefficients and/or other syntax information. Scanning techniques may be performed on the quantized transform coefficients in order to define one or more serialized one-dimensional vectors of coefficients from two-dimensional video blocks. The scanned coefficients are then entropy coded along with any syntax information, e.g., via content adaptive variable length coding (CAVLC), context adaptive binary arithmetic coding (CABAC), or another entropy coding process.

[0050] As part of the encoding process, encoded video blocks may be decoded in order to generate the video data that is used for subsequent prediction-based coding of subsequent video blocks. This is often referred to as a decoding loop of the encoding process, and generally mimics the decoding that is performed by a decoder device. In the decoding loop of an encoder or a decoder, filtering techniques may be used to improve video quality, and e.g., smooth pixel boundaries and possibly remove artifacts from decoded video. This filtering may be in-loop or post-loop. With in-loop filtering, the filtering of reconstructed video data occurs in the coding loop, which means that the filtered data is stored by an encoder or a decoder for subsequent use in the prediction of subsequent image data. In contrast, with post-loop filtering the filtering of reconstructed video data occurs out of the coding loop, which means that unfiltered versions of the data are stored by an encoder or a decoder for subsequent use in the prediction of subsequent image data. The loop filtering often follows a separate deblock filtering process, which typically applies filtering to pixels that are on or near boundaries of adjacent video blocks in order to remove blockiness artifacts that manifest at video block boundaries.

[0051] Relative to previous coding standards, the emerging HEVC standard introduces new terms and block sizes for video blocks. In particular, HEVC refers to coding units (CUs), which can be partitioned according to a quadtree partitioning scheme. An “LCU” refers to the largest sized coding unit (e.g., the “largest coding unit”) supported in a given situation. The LCU size may itself be signaled as part of the bitstream, e.g.,

as sequence level syntax. The LCU can be partitioned into smaller CUs. The CUs may be partitioned into prediction units (PUs) for purposes of prediction. The PUs may have square or rectangular shapes. Transforms are not fixed in the emerging HEVC standard, but are defined according to transform unit (TU) sizes, which may be the same size as a given CU, or possibly smaller. The residual data for a given CU may be communicated in the TUs. Syntax elements may be defined at the LCU level, the CU level, the PU level and the TU level.

[0052] To illustrate video blocks according to the HEVC standard, FIG. 2 conceptually shows an LCU of depth 64 by 64, which is then partitioned into smaller CUs according to a quadtree partitioning scheme. Elements called "split flags" may be included as CU-level syntax to indicate whether any given CU is itself sub-divided into four more CUs. In FIG. 2, CU₀ may comprise the LCU, CU₁ through CU₄ may comprise sub-CUs of the LCU. The bi-predictive merge mode syntax elements, as described in this disclosure, may be defined at the CU level (or possibly the LCU level if the LCU is not split into smaller CUs). Bi-predictive merge mode might also be supported for PUs of CUs, in some examples.

[0053] FIG. 3 further illustrates one possible relationship between CUs, PUs and TUs, which may be consistent with the emerging HEVC standard or other standards. However, other relationships are also possible, and FIG. 3 is merely shown as one possible example. In this case, any given CU of an LCU may itself be partitioned into PUs, such as those illustrated in FIG. 3. The PU type for a given CU may be signaled as CU-level syntax. As shown in FIG. 3, symmetric type PUs and asymmetric type PUs can be defined for a given CU. Moreover, two different TU structures may be defined for each of the four symmetric type PUs and asymmetric type PUs. Thus, a one-bit syntax element (TU size flag) may be used to signal the TU size, which may also be dependent on the PU type (symmetric or asymmetric). Coded block patterns (CBPs) may be defined for an LCU in order to indicate whether any given CU includes non-zero transform coefficients (e.g., whether any TUs are present).

[0054] In other examples, TUs may be defined differently than shown in FIG. 3. For example, residual samples corresponding to a CU may be subdivided into smaller units using a quadtree structure known as "residual quad tree" (RQT). The leaf nodes of the RQT may be referred as transform units (TUs). In some cases, TU's may be defined for a CU according to a quadtree structure, but the TU's may not necessarily depend on the PU's defined for any given CU. The PU's used for prediction may be defined

separately from TU's of any CU. A number of different types of partitioning schemes for CU's, TU's, and PU's are possible.

[0055] FIG. 4 is a block diagram illustrating a video encoder 50 consistent with this disclosure. Video encoder 50 may correspond to video encoder 22 of device 20, or a video encoder of a different device. As shown in FIG. 4, video encoder 50 includes a prediction encoding unit 32, quadtree partition unit 31, adders 48 and 51, and a memory 34. Video encoder 50 also includes a transform unit 38 and a quantization unit 40, as well as an inverse quantization unit 42 and an inverse transform unit 44. Video encoder 50 also includes an entropy coding unit 46, and a filter unit 47, which may include deblock filters and post loop and/or in loop filters. The encoded video data and syntax information that defines the manner of the encoding may be communicated to entropy encoding unit 46, which performs entropy encoding on the bitstream.

[0056] As shown in FIG. 4, prediction encoding unit 32 may support a plurality of different coding modes 35 in the encoding of video blocks. Modes 35 may include inter coding modes that define predictive data from different video frames (or slices). The inter coding modes may be bi-predictive, meaning that two different lists (e.g., List 0 and List 1) of predictive data (and typically two different motion vectors) are used to identify the predictive data. The inter coding modes may alternatively be uni-predictive, meaning that one list (e.g., List 0) of predictive data (and typically one motion vector) is used to identify the predictive data. Interpolations, offsets or other techniques may be performed in conjunction with the generation of predictive data. So-called SKIP modes and DIRECT modes may also be supported, which inherit motion information associated with a co-located block of another frame (or slice). SKIP mode blocks do not include any residual information, while DIRECT mode blocks include residual information.

[0057] In addition, modes 35 may include inter coding modes, which define predictive data based on data within the same video frame (or slice) as that being coded. Intra coding modes may include directional modes that define predictive data based on data in a particular direction within the same frame, as well as DC and/or planar modes that define predictive data based on the average or weighted average of neighboring data. Prediction encoding unit 32 may select the mode for a given block based on some criteria, such as based on an rate-distortion analysis or some characteristics of the block, such as the block size, texture or other characteristics.

[0058] In accordance with this disclosure, prediction encoding unit 32 supports a bi-predictive merge mode 35X. With bi-predictive merge mode 35X, a video block being coded inherits motion information from two different neighboring blocks, wherein the two different neighboring blocks were each encoded in a uni-predictive mode. Thus, the video block is coded with two different motion vectors that come from two different neighboring video blocks. In this case, prediction encoding unit 32 outputs an indication that the bi-predictive merge mode was used for a given block, and outputs syntax elements that identify the two different uni-predictive neighbors that collectively define the motion information for the current bi-predictive block. Predictive blocks associated with the bi-predictive merge mode may be combined into one bi-predictive block (possibly using weighting factors), and the bi-predictive block may be subtracted from the block being coded (via summer 48) to define residual data associated with the block coded in the bi-predictive merge mode.

[0059] The motion information may comprise two different uni-predictive motion vectors associated with the two different neighboring video blocks. These two different uni-predictive motion vectors can be used as the two bi-predictive motion vectors of the current video block. The motion information may further comprise two reference index values associated with the two different uni-predictive motion vectors, wherein the reference index values identify one or more lists of predictive data associated with the two different uni-predictive motion vectors. Again, residual data may be generated as the difference between the block being coded and the predictive data defined by the two different uni-predictive motion vectors that collectively define the bi-predictive merge block used in the prediction.

[0060] In the case of HEVC, the current video block being coded may comprise a so-called CU defined relative to an LCU according to a quadtree partitioning scheme. In this case, quadtree partition unit 31 may generate LCU syntax data that defines the quadtree partitioning scheme, and prediction encoding unit 32 may generate mode information for the CU that defines the bi-predictive merge mode, wherein the one or more syntax elements (which identify the two uni-predictive neighbors) are included in the mode information for the CU.

[0061] The described bi-predictive merge mode may increase the number of bi-predictive candidates that can be used in the context of merge mode coding. For example, if none of the neighbors are coded in bi-predictive mode, the described bi-predictive merge mode may allow bi-prediction to be exploited by combining the

motion information of two neighbors in predicting the current video block. Also, even if one or more neighbors are coded in bi-predictive mode, combining two uni-directional neighbors according to the described bi-predictive merge mode may still provide coding gains in some situations.

[0062] Generally, during the encoding process, video encoder 50 receives input video data. Prediction encoding unit 32 performs predictive coding techniques on video blocks (e.g. CUs and PUs). Quadtree partition unit 31 may break an LCU into smaller CU's and PU's according to HEVC partitioning explained above with reference to FIGS. 2 and 3. For inter coding, prediction encoding unit 32 compares CUs or PUs to various predictive candidates in one or more video reference frames or slices (e.g., one or more "list" of reference data) in order to define a predictive block. For intra coding, prediction encoding unit 32 generates a predictive block based on neighboring data within the same video frame or slice. Prediction encoding unit 32 outputs the prediction block and adder 48 subtracts the prediction block from the CU or PU being coded in order to generate a residual block. Again, at least some video blocks may be coded using the bi-predictive merge mode described herein.

[0063] FIG. 5 illustrates one example of prediction encoding unit 32 of video encoder 50 in greater detail. Prediction encoding unit 32 may include a mode selection unit 75 that selects the desired mode from modes 35, which include bi-predictive merge mode 35X as a possibility. For inter coding, prediction encoding unit 32 may comprise motion estimation (ME) unit 76 and motion compensation (MC) unit 77 that identify one or more motion vectors that points to predictive data, and generates the prediction block based on the motion vector. Typically, motion estimation is considered the process of generating the one or more motion vectors, which estimate motion. For example, the motion vector may indicate the displacement of a predictive block within a predictive frame relative to the current block being coded within the current frame. In the case bi-predictive merge mode 35X, two uni-directional motion vectors of two neighbors are combined to create a bi-directional prediction.

[0064] Motion compensation is typically considered the process of fetching or generating the predictive block (or blocks) based on the motion vector determined by motion estimation. In some cases, motion compensation for inter-coding may include interpolations to sub-pixel resolution, which permits the motion estimation process to estimate motion of video blocks to such sub-pixel resolution. Weighted combinations of two blocks (in the case of bi-prediction) may also be used.

[0065] For intra coding, prediction encoding unit 32 may comprise intra prediction unit 78. In this case, predictive data may be generated based on data within the current video block (e.g., adjacent to the video block being coded). Again, intra coding modes may include directional modes that define predictive data based on data in a particular direction within the same frame, as well as DC and/or planar modes that define predictive data based on the average or weighted average of neighboring data.

[0066] Rate-distortion (R-D) unit 79 may compare coding results of video blocks (e.g., CUs or PUs) in different modes. In addition R-D unit 79 may allow for other types of parameter adjustments, such as adjustments to interpolations, offsets, quantization parameters, or other factors that can affect the coding rate. Mode selection unit 75 can analyze the coding results in terms of coding rate (i.e., coding bits required for the block) and distortion (e.g., representing the video quality of the coded block relative to the original block) in order to make mode selections for video blocks. In this way, R-D unit 79 provides analysis of the results of different modes to allow mode selection unit 75 to select the desired mode for different video blocks. Consistent with this disclosure, the bi-predictive merge mode 35X may be selected when R-D unit 79 identifies it as the desired mode for a given video block, e.g., due to coding gains or coding efficiency.

[0067] Referring again to FIG. 4, after prediction encoding unit 32 outputs the prediction block, and after adder 48 subtracts the prediction block from the video block being coded in order to generate a residual block of residual pixel values, transform unit 38 applies a transform to the residual block. The transform may comprise a discrete cosine transform (DCT) or a conceptually similar transform such as that defined by the ITU H.264 standard or the HEVC standard. So-called “butterfly” structures may be defined to perform the transforms, or matrix-based multiplication could also be used. In some examples, consistent with the HEVC standard, the size of the transform may vary for different CUs, e.g., depending on the level of partitioning that occurs with respect to a given LCU. Transform units (TUs) may be defined in order to set the transform size applied by transform unit 38. Wavelet transforms, integer transforms, sub-band transforms or other types of transforms could also be used. In any case, transform unit applies the transform to the residual block, producing a block of residual transform coefficients. The transform, in general, may convert the residual information from a pixel domain to a frequency domain.

[0068] Quantization unit 40 then quantizes the residual transform coefficients to further reduce bit rate. Quantization unit 40, for example, may limit the number of bits used to

code each of the coefficients. In particular, quantization unit 40 may apply the delta QP defined for the LCU so as to define the level of quantization to apply (such as by combining the delta QP with the QP of the previous LCU or some other known QP). After quantization is performed on residual samples, entropy coding unit 46 may scan and entropy encode the data.

[0069] CAVLC is one type of entropy coding technique supported by the ITU H.264 standard and the emerging HEVC standard, which may be applied on a vectorized basis by entropy coding unit 46. CAVLC uses variable length coding (VLC) tables in a manner that effectively compresses serialized “runs” of coefficients and/or syntax elements. CABAC is another type of entropy coding technique supported by the ITU H.264 standard or the HEVC standard, which may be applied on a vectorized basis by entropy coding unit 46. CABAC may involve several stages, including binarization, context model selection, and binary arithmetic coding. In this case, entropy coding unit 46 codes coefficients and syntax elements according to CABAC. Many other types of entropy coding techniques also exist, and new entropy coding techniques will likely emerge in the future. This disclosure is not limited to any specific entropy coding technique.

[0070] Following the entropy coding by entropy encoding unit 46, the encoded video may be transmitted to another device or archived for later transmission or retrieval. The encoded video may comprise the entropy coded vectors and various syntax information (including the syntax information that defines two neighbors in the case of bi-predictive merge mode). Such information can be used by the decoder to properly configure the decoding process. Inverse quantization unit 42 and inverse transform unit 44 apply inverse quantization and inverse transform, respectively, to reconstruct the residual block in the pixel domain. Summer 51 adds the reconstructed residual block to the prediction block produced by prediction encoding unit 32 to produce a reconstructed video block for storage in memory 34. Prior to such storage, however, filter unit 47 may apply filtering to the video block to improve video quality. The filtering applied by filter unit 47 may reduce artifacts and smooth pixel boundaries. Moreover, filtering may improve compression by generating predictive video blocks that comprise close matches to video blocks being coded.

[0071] According to this disclosure, bi-predictive merge mode 35X that inherits motion information from two different neighboring blocks is supported, wherein the two different neighboring blocks were each encoded in a uni-predictive mode. The

described bi-predictive merge mode 35X may increase the number of bi-predictive candidates that can be used in the context of merge mode coding. Accordingly, R-D unit 79 (FIG. 5) may identify bi-predictive merge mode 35X as the most desirable coding mode due to coding gains achieved by this mode relative to other modes. In such cases, mode selection unit 75 may select bi-predictive merge mode 35X for coding one video blocks.

[0072] FIG. 6 is a block diagram illustrating an example of a video decoder 60, which decodes a video sequence that is encoded in the manner described herein. The techniques of this disclosure may be performed by video decoder 60 in some examples. In particular, video decoder 60 receives one or more syntax elements for a current video block, wherein the current video block is encoded according to a bi-predictive merge mode, and based on the one or more syntax elements, identifies two different neighboring video blocks coded in uni-predictive modes. Video decoder 60 then uses motion information of the two different neighboring video blocks to decode a current video block according to the bi-predictive merge mode.

[0073] A video sequence received at video decoder 60 may comprise an encoded set of image frames, a set of frame slices, a commonly coded group of pictures (GOPs), or a wide variety of units of video information that include encoded LCUs (or other video blocks) and syntax information to define how to decode such LCUs. The process of decoding the LCUs may include decoding an indication of the coding mode, which may be the bi-predictive merge mode described herein.

[0074] Video decoder 60 includes an entropy decoding unit 52, which performs the reciprocal decoding function of the encoding performed by entropy encoding unit 46 of FIG. 2. In particular, entropy decoding unit 52 may perform CAVLC or CABAC decoding, or any other type of entropy decoding used by video encoder 50. Video decoder 60 also includes a prediction decoding unit 54, an inverse quantization unit 56, an inverse transform unit 58, a memory 62, and a summer 64. In particular, like video encoder 50, video decoder 60 includes a prediction decoding unit 54 and a filter unit 57. Prediction decoding unit 54 of video decoder 60 may include motion compensation unit 86, which decodes inter coded blocks and possibly includes one or more interpolation filters for sub-pixel interpolation in the motion compensation process. Prediction decoding unit 54 may also include an intra prediction unit for decoding intra modes. Prediction decoding unit 54 may support a plurality of modes 35 including bi-predictive merge mode 55X. Filter unit 57 may filter the output of summer 64, and may receive

entropy decoded filter information so as to define the filter coefficients applied in the loop filtering.

[0075] Upon receiving encoded video data, entropy decoding unit 52 performs reciprocal decoding to the encoding performed by entropy encoding unit 46 (of encoder 50 in FIG. 4). At the decoder, entropy decoding unit 52 parses the bitstream to determine LCU's and the corresponding partitioning associated with the LCU's. In some examples, an LCU or the CUs of the LCU may define coding modes that were used, and these coding modes may include the bi-predictive merge mode. Accordingly, entropy decoding unit 52 may forward the syntax information to prediction unit that identifies the bi-predictive merge mode. In this case, the syntax information may include one or more syntax elements that identify two different neighboring video blocks coded in uni-predictive modes. In this case, MC unit 86 of prediction decoding unit 54 may use motion information of the two different neighboring video blocks to decode a current video block according to the bi-predictive merge mode. That is, MC unit 86 may fetch predictive data identified by the motion information of the two different neighboring video blocks, and use some combination of this predictive data in decoding the current video block in bi-predictive merge mode.

[0076] FIG. 7 is a conceptual illustration showing one example of five different neighbors that may be considered for purposes of bi-predictive merge mode. In this example, the top (T) neighbor, top right (TR) neighbor, left (L) neighbor, bottom left (BL) neighbor, and co-located temporal (Temp) neighbor from another video frame may be considered for purposes of bi-predictive merge mode. Of course, other neighbors (spatial or temporal) could also be used for any merge mode inheritance of motion information.

[0077] Again, with merge mode, the current video block may inherit all motion information of a neighboring candidate block. That means the current block will have the same motion vector, same reference frame, and same prediction mode (uni-prediction or bi-prediction) as the selected neighboring block. The selected neighboring block may be signaled as part of an encoded bitstream, but the motion information does not need to be signaled because the decoder can obtain the motion information from the selected neighboring block.

[0078] Consistent with this disclosure, a bi-predictive merge mode is supported, which inherits motion information from two different neighboring blocks, wherein the two different neighboring blocks were each encoded in a uni-predictive mode. The

described bi-predictive merge mode may increase the number of bi-predictive candidates that can be used in the context of merge mode. Rather than signaling one neighbor, the bi-predictive merge mode may signal two different neighbors. The bi-predictive merge mode may be an extension to a conventional merge mode by simply increasing candidate neighboring blocks to include combinations thereof, or may be an entirely separate mode relative to a conventional merge mode.

[0079] Assuming the spatial and temporal candidate neighboring blocks shown in FIG. 1, the described bi-predictive merge mode may work in at least two scenarios. In a first scenario, all neighboring candidate neighboring blocks are coded in uni-predictive modes. In this case, any two of the candidate blocks can be selected, and the motion information from both selected candidates can be combined to achieve a bi-prediction. For example, assume that the neighboring blocks shown in FIG. 1 are coded according to the following information:

L: uni-pred, L0, refIdx = 0
T: uni-pred, L1, refIdx = 0
TR: uni-pred, L0, refIdx = 1
BL: uni-pred, L0, refIdx = 0
Temp: uni-pred, L1, refIdx = 1

In this case, there are 10 combinations of any two of five candidates. L0 refers to a first list of predictive data, and L1 refers to a second list of predictive data. The refIdx may comprise an index to a particular picture in the respective list. The video encoder can select the best combination (e.g., in terms of encoding rate and distortion), and can send syntax information that identifies the two selected neighboring blocks. The decoder can decode the syntax information and obtain the motion information from the selected neighboring blocks.

[0080] In a second scenario, at least one of the candidate neighboring blocks may be coded in a bi-predictive mode. In this case, any two of the uni-predictive neighboring blocks can be considered (in combination) to obtain a bi-predictive merge mode candidate. However, any bi-predictive blocks can also be used alone for consideration as a bi-predictive merge candidate. For example, assume that the neighboring blocks shown in FIG. 1 are coded according to the following information:

L: bi-pred, L0, refIdx = 0, L1, refIdx = 0
T: uni-pred, L1, refIdx = 0
TR: uni-pred, L0, refIdx = 1

BL: uni-pred, L0, refIdx = 0

Temp: bi-pred, L0, refIdx=0, L1, refIdx = 1

Again, L0 may comprise a value that refers to a first list of predictive data, L1 may comprise a value that refers to a second list of predictive data, and refIdx may be values that define indexes to a particular picture in the respective list. In this second example, two of the five candidates are already bi-predictive mode, so these can be considered alone for purposes of bi-predictive merge modes. In addition, different combinations of the three remaining uni-predictive candidates can be considered. So, in this case, there will be 5 bi-predictive mode possibilities:

1. L
2. Temp
3. T + TR
4. T + BL
5. TR + BL

[0081] In this second example, the encoder can select the best neighbor (or combination of neighbors) from these five possibilities (e.g., in terms of encoding rate and distortion), and can send syntax information that identifies which individual neighbor or combination of neighbors was used in the merge mode. The decoder can decode the syntax information and obtain the motion information from the selected neighboring block(s).

[0082] In the example of FIG. 7, five candidates are shown. However, additional candidates may also be considered either in the same areas as the candidates of FIG. 7 or in other areas. In some cases, there may be several top (T) candidates, several top left (TL) candidates, several left (L) candidates, several bottom left (BL) candidates, and several temporal (T) candidates. In some cases, the size of the current block may differ from that of the candidates, in which case, the top edge or left edge of the current block may be adjacent several candidates. In other cases, candidates at even further distances from the current video block may be considered for purposes of the bi-predictive merge mode described in this disclosure. Many different scenarios using many different candidates are possible consistent with this disclosure. Thus, FIG. 7 is merely one example illustrating five neighboring candidates relative to the current video block.

[0083] FIG. 8 is a flow diagram illustrating a decoding technique consistent with this disclosure. FIG. 8 will be described from the perspective of video decoder 60 of FIG. 6,

although other devices may perform similar techniques. As shown in FIG. 8, video decoder 60 receives an LCU including syntax elements for the LCU and for the CU's within the LCU (801). In particular, entropy decoding unit 52 may receive a bitstream that includes the LCU and parse the bitstream to identify the syntax elements, which can be forwarded to prediction decoding unit 54. Accordingly, prediction decoding unit 54 can modes of the CU's based on the syntax elements. In other examples, modes may be defined at the PU level rather than at the CU level.

[0084] In determining the modes, prediction decoding unit 54 identifies any CU's coded in bi-predictive merge mode (803). If any CU is not coded in bi-predictive merge mode ("no" 803), then that CU is decoded according to its mode (804). For example, many different intra modes and many different inter modes may be supported. If any CU is coded in the bi-predictive merge mode ("yes" 803), then this CU is bi-predictive. However, its motion vectors for bi-prediction come from two uni-directional neighbors as discussed herein. In this case, MC unit 86 of prediction decoding unit 54 identifies two different uni-predictive neighbors to the CU based on the CU syntax element (805), and uses motion information of the uni-predictive neighbors to decode the bi-predictive CU.

[0085] FIG. 9 is a flow diagram illustrating an encoding technique consistent with this disclosure. FIG. 9 will be described from the perspective of video encoder 50 of FIG. 4, although other devices may perform similar techniques. As shown in FIG. 9, prediction encoding unit 32 selects a bi-predictive merge mode for a CU. For example, prediction encoding unit 32 (see FIG. 5) may include a mode selection unit 75 that selects the bi-predictive merge mode 35X for a CU based on a plurality of possible modes 35. R-D unit 75 may identify the coding rate and level of quality or distortion associated with different modes by analyzing the results of coding by ME unit 76 and MC unit 77 for various inter modes and by analyzing the results of coding by intra prediction unit 78 for intra modes. In this way, mode selection unit 75 may identify the best mode for any given situation.

[0086] Once prediction encoding unit 32 selects the bi-predictive merge mode 35X for a CU, prediction unit identifies two different neighboring blocks coded in uni-predictive modes (902). This process of identifying the two different neighboring blocks coded in uni-predictive modes may be performed by ME unit 76 in the manner similar to that described above. For example, ME unit 76 and MC unit 77 may generate coding results of different combinations of motion information from different uni-directional

neighbors, and these results may be analyzed by R-D unit 75 to determine the coding rate and quality or distortion associated with such different combinations. Ultimately, R-D unit 75 can determine which bi-predictive merge mode combination of uni-predictive neighbors yields the best coding results.

[0087] Accordingly, prediction encoding unit 32 uses the motion information of the best combination of two uni-predictive neighbors to encoding the CU as bi-predictive (903). Of course, any bi-predictive neighbors may also be considered, and possibly used for merge mode coding if the rate-distortion results are better than using two uni-predictive neighbors. Prediction unit 32 (e.g., ME unit 76 or MC unit 77) generates one or more syntax elements for the CU to identify the two different uni-predictive neighbors used for the bi-predictive merge mode coding of the CU. The syntax elements, for example, may comprise index values that identify two of the neighbors of the CU, such as the left (L) neighbor, bottom left (BL) neighbor, (T) top neighbor, top right (TR) neighbor, or co-located temporal (T) neighbor as conceptually illustrated in FIG. 7. However, many other signaling schemes for the syntax elements may also be used.

[0088] The techniques of this disclosure may be realized in a wide variety of devices or apparatuses, including a wireless handset, and integrated circuit (IC) or a set of ICs (i.e., a chip set). Any components, modules or units have been described provided to emphasize functional aspects and does not necessarily require realization by different hardware units.

[0089] Accordingly, the techniques described herein may be implemented in hardware, software, firmware, or any combination thereof. Any features described as modules or components may be implemented together in an integrated logic device or separately as discrete but interoperable logic devices. If implemented in software, the techniques may be realized at least in part by a computer-readable medium comprising instructions that, when executed, performs one or more of the methods described above. The computer-readable data storage medium may form part of a computer program product, which may include packaging materials.

[0090] The computer-readable medium may comprise a tangible computer readable storage medium, such as random access memory (RAM) such as synchronous dynamic random access memory (SDRAM), read-only memory (ROM), non-volatile random access memory (NVRAM), electrically erasable programmable read-only memory (EEPROM), FLASH memory, magnetic or optical data storage media, and the like. The techniques additionally, or alternatively, may be realized at least in part by a computer-

readable communication medium that carries or communicates code in the form of instructions or data structures and that can be accessed, read, and/or executed by a computer.

[0091] The instructions may be executed by one or more processors, such as one or more digital signal processors (DSPs), general purpose microprocessors, application specific integrated circuits (ASICs), field programmable logic arrays (FPGAs), or other equivalent integrated or discrete logic circuitry. The term “processor,” as used herein may refer to any of the foregoing structure or any other structure suitable for implementation of the techniques described herein. In addition, in some aspects, the functionality described herein may be provided within dedicated software modules or hardware modules configured for encoding and decoding, or incorporated in a combined video encoder-decoder (CODEC). Also, the techniques could be fully implemented in one or more circuits or logic elements.

[0092] Various aspects of the disclosure have been described. These and other aspects are within the scope of the following claims.

CLAIMS

1. A method of decoding video data, the method comprising:
 - receiving one or more syntax elements for a current video block, wherein the one or more syntax elements define one or more merge mode candidates from a set of candidates to be used to code the current video block according to a merge mode, wherein a first candidate in the set of candidates is a left candidate, a second candidate in the set of candidates is a temporal candidate, and a third candidate in the set of candidates is a bi-predictive merge mode candidate defined according to two different neighboring video blocks that are coded in uni- predictive modes; and
 - when the one or more syntax elements define the third candidate,
 - identifying the two different neighboring video blocks coded in the uni- predictive modes based on the one or more syntax elements; and
 - using motion information of the two different neighboring video blocks to decode the current video block.
2. The method of claim 1, wherein the motion information comprises two different uni-predictive motion vectors associated with the two different neighboring video blocks.
3. The method of claim 2, wherein the motion information further comprises at least two values associated with the two different uni-predictive motion vectors, wherein the values identify one or more lists of predictive data associated with the two different uni-predictive motion vectors.
4. The method of claim 1, wherein the current video block comprises a coding unit (CU) defined according to a high efficiency video coding (HEVC) standard, wherein the CU is defined relative to a largest coding unit (LCU) according to a quadtree partitioning scheme, the method further comprising:
 - receiving LCU syntax data that defines the quadtree partitioning scheme; and
 - receiving mode information for the CU that defines the bi-predictive merge mode candidate, wherein the one or more syntax elements are included in the mode information for the CU.

5. The method of claim 1, wherein the current video block comprises a prediction unit (PU) of a coding unit (CU) that is defined according to a high efficiency video coding (HEVC) standard.

6. A method of encoding video data, the method comprising:
selecting a candidate from a set of candidates for encoding a current video block according to a merge mode, wherein a first candidate in the set of candidates is a left candidate, a second candidate in the set of candidates is a temporal candidate, and a third candidate in the set of candidates is a bi-predictive merge mode candidate defined according to two different neighboring video blocks that are coded in the uni- predictive modes; and
upon selecting the third candidate:
identifying the two different neighboring video blocks coded in the uni- predictive modes;
using motion information of the two different neighboring video blocks to encode the current video block; and
generating one or more syntax elements that identify the two different neighboring video blocks that are coded in uni- predictive modes to a video decoder.

7. The method of claim 6, wherein the motion information comprises two different uni-predictive motion vectors associated with the two different neighboring video blocks.

8. The method of claim 7, wherein the motion information further comprises at least two values associated with the two different uni-predictive motion vectors, wherein the values identify one or more lists of predictive data associated with the two different uni-predictive motion vectors.

9. The method of claim 6, wherein the current video block comprises a coding unit (CU) defined according to a high efficiency video coding (HEVC) standard, the method further comprising:
defining the CU relative to a largest coding unit (LCU) according to a quadtree partitioning scheme;

generating an LCU syntax data that defines the quadtree partitioning scheme;
and

generating mode information for the CU that defines the bi-predictive merge mode candidate, wherein the one or more syntax elements are included in the mode information for the CU.

10. The method of claim 6, wherein the current video block comprises a prediction unit (PU) of a coding unit (CU) that is defined according to a high efficiency video coding (HEVC) standard.

11. A video decoding device that decodes video data, the video decoding device comprising:

a video decoder configured to:

receive one or more syntax elements for a current video block, wherein the one or more syntax elements define one or more merge mode candidates from a set of candidates to be used to code the current video block according to a merge mode, wherein a first candidate in the set of candidates is a left candidate, a second candidate in the set of candidates is a temporal candidate, and a third candidate in the set of candidates is a bi-predictive merge mode candidate defined according to two different neighboring video blocks that are coded in uni- predictive modes; and

when the one or more syntax elements define the third candidate,

identify the two different neighboring video blocks coded in the uni-predictive modes based on the one or more syntax elements; and

use motion information of the two different neighboring video blocks to decode the current video block.

12. The video decoding device of claim 11, wherein the video decoder includes:

an entropy decoding unit configured to receive and decode the one or more syntax elements; and

a prediction unit configured to identify the two different neighboring video blocks, and use the motion information of the two different neighboring video blocks to decode the current video block.

13. The video decoding device of claim 11, wherein the motion information comprises two different uni-predictive motion vectors associated with the two different neighboring video blocks.

14. The video decoding device of claim 13, wherein the motion information further comprises at least two values associated with the two different uni-predictive motion vectors, wherein the values identify one or more lists of predictive data associated with the two different uni-predictive motion vectors.

15. The video decoding device of claim 11, wherein the current video block comprises a coding unit (CU) defined according to a high efficiency video coding (HEVC) standard, wherein the CU is defined relative to a largest coding unit (LCU) according to a quadtree partitioning scheme, wherein the decoder:

- receives LCU syntax data that defines the quadtree partitioning scheme; and
- receives mode information for the CU that defines the bi-predictive merge mode candidate, wherein the one or more syntax elements are included in the mode information for the CU.

16. The video decoding device of claim 11, wherein the current video block comprises a prediction unit (PU) of a coding unit (CU) that is defined according to a high efficiency video coding (HEVC) standard.

17. The video decoding device of claim 11, wherein the video decoding device comprises one or more of:

- an integrated circuit;
- a microprocessor; and
- a wireless communication device that includes a video encoder.

18. A video encoding device comprising:

- a video encode configured to:
 - select a candidate from a set of candidates for encoding a current video block according to a merge mode, wherein a first candidate in the set of candidates is a left candidate, a second candidate in the set of candidates is a temporal candidate, and a

third candidate in the set of candidates is a bi-predictive merge mode candidate defined according to two different neighboring video blocks that are coded in uni- predictive modes; and

upon selecting the third candidate:

identify the two different neighboring video blocks coded in the uni-predictive modes;

uses motion information of the two different neighboring video blocks to encode the current video block; and

generates one or more syntax elements that identify the two different neighboring video blocks that are coded in the uni- predictive modes to a video decoder.

19. The video encoding device of claim 18, wherein the motion information comprises two different uni-predictive motion vectors associated with the two different neighboring video blocks.

20. The video encoding device of claim 19, wherein the motion information further comprises at least two values associated with the two different uni-predictive motion vectors, wherein the values identify one or more lists of predictive data associated with the two different uni-predictive motion vectors.

21. The video encoding device of claim 18, wherein the current video block comprises a coding unit (CU) defined according to a high efficiency video coding (HEVC) standard, wherein the video encoder includes:

a quadtree partition unit configured to define the CU relative to a largest coding unit (LCU) according to a quadtree partitioning scheme, and generate an LCU syntax data that defines the quadtree partitioning scheme; and

a prediction unit configured to generate mode information for the CU that defines the bi-predictive merge mode candidate, wherein the one or more syntax elements are included in the mode information for the CU.

22. The video encoding device of claim 18, wherein the current video block comprises a prediction unit (PU) of a coding unit (CU) that is defined according to a high efficiency video coding (HEVC) standard.

23. The video encoding device of claim 18, wherein the video encoding device comprises one or more of:

- an integrated circuit;
- a microprocessor; and
- a wireless communication device that includes a video encoder.

24. A device for decoding video data, the device comprising:

- means for receiving one or more syntax elements for a current video block, wherein the one or more syntax elements define one or more merge mode candidates from a set of candidates to be used to code the current video block according to a merge mode, wherein a first candidate in the set of candidates is a left candidate, a second candidate in the set of candidates is a temporal candidate, and a third candidate in the set of candidates is a bi-predictive merge mode candidate defined according to two different neighboring video blocks that are coded in uni- predictive modes; and
 - when the one or more syntax elements define the third candidate,
 - means for identifying two the different neighboring video blocks coded in the uni-predictive modes based on the one or more syntax elements; and
 - means for using motion information of the two different neighboring video blocks to decode the current video block.

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

- means for selecting a candidate from a set of candidates for encoding a current video block according to a merge mode, wherein a first candidate in the set of candidates is a left candidate, a second candidate in the set of candidates is a temporal candidate, and a third candidate in the set of candidates is a bi-predictive merge mode candidate defined according to two different neighboring video blocks that are coded in uni- predictive modes; and
 - upon selecting the third candidate:
 - means for identifying the two different neighboring video blocks coded in the uni-predictive modes;
 - means for using motion information of the two different neighboring video blocks to encode the current video block; and

means for generating one or more syntax elements that identify the two different neighboring video blocks that are coded in the uni- predictive modes to a video decoder.

26. A computer-readable medium comprising instructions that upon execution cause a processor to decode video data, wherein the instructions cause the processor to:

upon receiving one or more syntax elements for a current video block, wherein the one or more syntax elements define one or more merge mode candidates from a set of candidates to be used to code the current video block according to a merge mode, wherein a first candidate in the set of candidates is a left candidate, a second candidate in the set of candidates is a temporal candidate, and a third candidate in the set of candidates is a bi-predictive merge mode candidate defined according to two different neighboring video blocks that are coded in uni- predictive modes,

when the one or more syntax elements define the third candidate,

identify the two different neighboring video blocks coded in the uni-predictive modes based on the one or more syntax elements; and

use motion information of the two different neighboring video blocks to decode the current video block.

27. A computer-readable medium comprising instructions that upon execution cause a processor to encode video data, wherein the instructions cause the processor to:

select a candidate from a set of candidates for encoding a current video block according to a merge mode, wherein a first candidate in the set of candidates is a left candidate, a second candidate in the set of candidates is a temporal candidate, and a third candidate in the set of candidates is a bi-predictive merge mode candidate defined according to two different neighboring video blocks that are coded in uni- predictive modes; and

upon selecting the third candidate:

identify the two different neighboring video blocks coded in the uni-predictive modes;

use motion information of the two different neighboring video blocks to encode the current video block; and

generate one or more syntax elements that identify the two different neighboring video blocks that are coded in the uni- predictive modes to a video decoder.