

CONSTRAINTS ON QUANTIZED RESIDUAL DIFFERENTIAL PULSE CODE MODULATION REPRESENTATION OF CODED VIDEO

CROSS REFERENCE TO RELATED APPLICATIONS

[001] This application is based on International Patent Application No. PCT/US2020/029603, filed on April 23, 2020, which claims the priority to and benefits of International Patent Application No. PCT/CN2019/084008, filed on April 24, 2019. All the aforementioned patent applications are hereby incorporated by reference in their entireties.

TECHNICAL FIELD

[002] This patent document relates to video coding techniques, devices and systems.

BACKGROUND

[003] In spite of the advances in video compression, digital video still accounts for the largest bandwidth use on the internet and other digital communication networks. As the number of connected user devices capable of receiving and displaying video increases, it is expected that the bandwidth demand for digital video usage will continue to grow.

SUMMARY

[004] The present document describes various embodiments and techniques in which a secondary transform is used during decoding or encoding of video or images.

[005] A first example method of video processing includes determining, based on an applicability rule, that a differential coding mode is applicable to a conversion between a chroma video block of a video and a bitstream representation of the chroma video block, and performing, based on the determining, the conversion between the chroma video block and the bitstream representation of the chroma video block using the differential coding mode, where, in the differential coding mode, the chroma video block is represented in the bitstream representation using a difference between a quantized residual of an intra prediction of the chroma video block and a prediction of the quantized residual, where the intra prediction is performed in a first

direction and the prediction of the quantized residual is performed in a second direction, and where, in the differential coding mode, the difference between the quantized residual and the prediction of the quantized residual is represented using a differential pulse coding modulation (DPCM) representation.

[006] A second example method of video processing includes performing a conversion between a current video block of a video and a bitstream representation of the current video block using a differential coding mode and using an intra prediction mode based on a coexistence rule, where, in the intra prediction mode, a quantized residual of an intra prediction of the current video block is based on predictions of samples for the current video block, where, in the differential coding mode, the current video block is represented in the bitstream representing using a difference between the quantized residual and the prediction of the quantized residual, and where the difference between the quantized residual and the prediction of the quantized residual is represented using a differential pulse coding modulation (DPCM) representation.

[007] A third example method of video processing includes performing a conversion between a current video block of a video and a bitstream representation of the current video block using a differential coding mode in which the current video block is represented in the bitstream representation using a difference between a quantized residual of an intra prediction of the current video block and a prediction of the quantized residual, where a direction of the prediction of the quantized residual is inferable from the bitstream representation.

[008] A fourth example method of video processing includes determining that a differential coding mode is applicable to a conversion between a current video block of a video and a bitstream representation of the current video block, and performing, based on the determining, the conversion between the current video block and the bitstream representation of the current video block using the differential coding mode according to an implementation rule, where, in the differential coding mode, the current video block is represented in the bitstream representation using a difference between a quantized residual of an intra prediction of the current video block and a prediction of the quantized residual, and where, in the differential coding mode, the difference between the quantized residual and the prediction of the quantized residual is represented using a differential pulse coding modulation (DPCM) representation.

[009] A fifth example method of video processing includes determining, based on an

applicability rule, that a differential coding mode is applicable to a conversion between a current video block of a video and a bitstream representation of the current video block, where the current video block is associated with a color component, and where the applicability rule specifies that whether the differential coding mode is enabled or disabled for the current video block based on whether the differential coding mode is applied to another video block of the video, and performing, based on the determining, the conversion between the current video block and the bitstream representation of the current video block using the differential coding mode, where, in the differential coding mode, the current video block is represented in the bitstream representation using a difference between a quantized residual of an intra prediction of the current video block and a prediction of the quantized residual, and where, in the differential coding mode, the difference between the quantized residual and the prediction of the quantized residual is represented using a differential pulse coding modulation (DPCM) representation.

[0010] In one example aspect, a video processing method is disclosed. The method includes performing a conversion between a current video block and a bitstream representation of the current video block using a differential coding mode and selectively using an intra prediction mode based on a coexistence rule; wherein the intra prediction mode is used for generating predictions for samples of the current video block; and wherein, the differential coding mode is used to represent a quantized residual block from the predictions of the pixels, using a differential pulse coding modulation representation.

[0011] In another example aspect, another method of video processing is disclosed. The method includes performing a conversion between a current video block and a bitstream representation of the current video block using a differential coding mode in which a quantized residual block from a predictions of pixels of the current video block is represented using a differential pulse coding modulation representation; wherein a first direction of the prediction or a second direction of the differential coding mode is inferable from the bitstream representation.

[0012] In yet another example aspect, another method of video processing is disclosed. The method includes determining, based on an applicability rule, that a differential coding mode is applicable to a conversion between a current video block and a bitstream representation of the current video block; and performing the conversion between a current video block and a bitstream representation using the differential coding mode; wherein, in the differential coding

mode, a quantized residual block from intra prediction of pixels of the current video block is represented using a differential pulse coding modulation representation performed in a residual prediction direction that is different from a horizontal or a vertical direction.

[0013] In yet another example aspect, another method of video processing is disclosed. The method includes determining that a differential coding mode is applicable to a conversion between a current video block and a bitstream representation of the current video block; and performing the conversion between a current video block and a bitstream representation using an implementation rule of the differential coding mode; wherein, in the differential coding mode, a quantized residual block from intra prediction of pixels of the current video block is represented using a differential pulse coding modulation representation performed in a residual prediction direction that is different from a horizontal or a vertical direction.

[0014] In yet another example aspect, a video processing apparatus is disclosed. The apparatus includes a processor configured to perform an above disclosed method.

[0015] In yet another example aspect, a computer readable medium is disclosed. The medium has code for processor-implementation of the above-described methods stored on it.

[0016] These, and other, aspects are described in the present document.

BRIEF DESCRIPTION OF THE DRAWINGS

[0017] FIG. 1 is an illustration of Intra block copy.

[0018] FIG. 2 shows an example of a block coded in palette mode.

[0019] FIG. 3 shows an example use of palette predictor to signal palette entries.

[0020] FIG. 4 shows examples of horizontal and vertical traverse scans.

[0021] FIG. 5 shows an example of coding of palette indices.

[0022] FIG. 6 shows an example process of affine linear weighted intra prediction (ALWIP) process.

[0023] FIG. 7 shows an example process of affine linear weighted intra prediction (ALWIP) process.

[0024] FIG. 8 shows an example process of affine linear weighted intra prediction (ALWIP) process.

[0025] FIG. 9 shows an example process of affine linear weighted intra prediction (ALWIP) process.

- [0026] FIG. 10 is a block diagram of an example hardware platform for implementing techniques described in the present document.
- [0027] FIG. 11 is a flowchart for an example method of video processing.
- [0028] FIG. 12 shows an example of four merge candidates.
- [0029] FIG. 13 shows example pairs of merge candidates used in video coding.
- [0030] FIG. 14 is a block diagram showing an example video processing system in which various techniques disclosed herein may be implemented.
- [0031] FIG. 15 is a block diagram that illustrates an example video coding system that may utilize the techniques of this disclosure.
- [0032] FIG. 16 is a block diagram illustrating an example of video encoder.
- [0033] FIG. 17 is a block diagram illustrating an example of video decoder.
- [0034] FIGS. 18-22 show five example flowcharts for example methods for video processing.

DETAILED DESCRIPTION

[0035] Section headings are used in the present document to facilitate ease of understanding and do not limit the embodiments disclosed in a section to only that section. Furthermore, while certain embodiments are described with reference to Versatile Video Coding or other specific video codecs, the disclosed techniques are applicable to other video coding technologies also. Furthermore, while some embodiments describe video coding steps in detail, it will be understood that corresponding steps decoding that undo the coding will be implemented by a decoder. Furthermore, the term video processing encompasses video coding or compression, video decoding or decompression and video transcoding in which video pixels are represented from one compressed format into another compressed format or at a different compressed bitrate.

1. Summary

[0036] This patent document is related to video coding technologies. Specifically, it is related to DPCM coding in video coding. It may be applied to the existing video coding standard like HEVC, or the standard (Versatile Video Coding) to be finalized. It may be also applicable to future video coding standards or video codec.

2. Initial discussion

[0037] Video coding standards have evolved primarily through the development of the well-

known ITU-T and ISO/IEC standards. The ITU-T produced H.261 and H.263, ISO/IEC produced MPEG-1 and MPEG-4 Visual, and the two organizations jointly produced the H.262/MPEG-2 Video and H.264/MPEG-4 Advanced Video Coding (AVC) and H.265/HEVC [1] standards. Since H.262, the video coding standards are based on the hybrid video coding structure wherein temporal prediction plus transform coding are utilized. To explore the future video coding technologies beyond HEVC, Joint Video Exploration Team (JVET) was founded by VCEG and MPEG jointly in 2015. Since then, many new methods have been adopted by JVET and put into the reference software named Joint Exploration Model (JEM) [3,4]. In April 2018, the Joint Video Expert Team (JVET) between VCEG (Q6/16) and ISO/IEC JTC1 SC29/WG11 (MPEG) was created to work on the VVC standard targeting at 50% bitrate reduction compared to HEVC.

[0038] The latest version of VVC draft, i.e., Versatile Video Coding (Draft 4) could be found at:

phenix.it-sudparis.eu/jvet/doc_end_user/current_document.php?id=5755

The latest reference software of VVC, named VTM, could be found at:

vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/tags/VTM-4.0

2.1 Intra block copy

[0039] Intra block copy (IBC), a.k.a. current picture referencing, has been adopted in HEVC Screen Content Coding extensions (HEVC-SCC) [1] and the current VVC test model (VTM-4.0). IBC extends the concept of motion compensation from inter-frame coding to intra-frame coding. As demonstrated in FIG. 1, the current block is predicted by a reference block in the same picture when IBC is applied. The samples in the reference block must have been already reconstructed before the current block is coded or decoded. Although IBC is not so efficient for most camera-captured sequences, it shows significant coding gains for screen content. The reason is that there are lots of repeating patterns, such as icons and text characters in a screen content picture. IBC can remove the redundancy between these repeating patterns effectively. In HEVC-SCC, an inter-coded coding unit (CU) can apply IBC if it chooses the current picture as its reference picture. The MV is renamed as block vector (BV) in this case, and a BV always has an integer-pixel precision. To be compatible with main profile HEVC, the current picture is marked as a “long-term” reference picture in the Decoded Picture Buffer (DPB). It should be

noted that similarly, in multiple view/3D video coding standards, the inter-view reference picture is also marked as a “long-term” reference picture.

[0040] Following a BV to find its reference block, the prediction can be generated by copying the reference block. The residual can be got by subtracting the reference pixels from the original signals. Then transform and quantization can be applied as in other coding modes.

[0041] FIG. 1 is an illustration of Intra block copy.

[0042] However, when a reference block is outside of the picture, or overlaps with the current block, or outside of the reconstructed area, or outside of the valid area restricted by some constraints, part or all pixel values are not defined. Basically, there are two solutions to handle such a problem. One is to disallow such a situation, e.g. in bitstream conformance. The other is to apply padding for those undefined pixel values. The following sub-sessions describe the solutions in detail.

2.2 IBC in HEVC Screen Content Coding extensions

[0043] In the screen content coding extensions of HEVC, when a block uses current picture as reference, it should guarantee that the whole reference block is within the available reconstructed area, as indicated in the following spec text:

[0044] The variables offsetX and offsetY are derived as follows: offsetX =

$(\text{ChromaArrayType} == 0) ? 0 : (\text{mvCLX}[0] \& 0x7 ? 2 : 0) \quad (0-1)$ offsetY =

$(\text{ChromaArrayType} == 0) ? 0 : (\text{mvCLX}[1] \& 0x7 ? 2 : 0) \quad (0-2)$

[0045] It is a requirement of bitstream conformance that when the reference picture is the current picture, the luma motion vector mvLX shall obey the following constraints:

- When the derivation process for z-scan order block availability as specified in clause 6.4.1 is invoked with (xCurr, yCurr) set equal to (xCb, yCb) and the neighbouring luma location (xNbY, yNbY) set equal to (xPb + (mvLX[0] >> 2) – offsetX, yPb + (mvLX[1] >> 2) – offsetY) as inputs, the output shall be equal to TRUE.
- When the derivation process for z-scan order block availability as specified in clause 6.4.1 is invoked with (xCurr, yCurr) set equal to (xCb, yCb) and the neighbouring luma location (xNbY, yNbY) set equal to (xPb + (mvLX[0] >> 2) + nPbW – 1 + offsetX, yPb + (mvLX[1] >> 2) + nPbH – 1 + offsetY) as inputs, the output shall be equal to TRUE.
- One or both the following conditions shall be true:
 - The value of (mvLX[0] >> 2) + nPbW + xB1 + offsetX is less than or equal to 0.
 - The value of (mvLX[1] >> 2) + nPbH + yB1 + offsetY is less than or equal to 0.

- The following condition shall be true:

$$\begin{aligned} & (xPb + (mvLX[0] \gg 2) + nPbSw - 1 + offsetX) / CtbSizeY - xCurr / CtbSizeY \leq \\ & yCurr / CtbSizeY - (yPb + (mvLX[1] \gg 2) + nPbSh - 1 + offsetY) / CtbSizeY \\ & (0-3) \end{aligned}$$

[0046] Thus, the case that the reference block overlaps with the current block or the reference block is outside of the picture will not happen. There is no need to pad the reference or prediction block.

2.3 IBC in VVC Test Model

[0047] In the current VVC test model, i.e. VTM-4.0 design, the whole reference block should be with the current coding tree unit (CTU) and does not overlap with the current block. Thus, there is no need to pad the reference or prediction block. The IBC flag is coded as a prediction mode of the current CU. Thus, there are totally three prediction modes, MODE_INTRA, MODE_INTER and MODE_IBC for each CU.

2.3.1 IBC Merge mode

[0048] In IBC merge mode, an index pointing to an entry in the IBC *merge candidates list* is parsed from the bitstream. The construction of the IBC merge list can be summarized according to the following sequence of steps:

- Step 1: Derivation of spatial candidates
- Step 2: Insertion of HMVP candidates
- Step 3: Insertion of pairwise average candidates

[0049] In the derivation of spatial merge candidates, a maximum of four merge candidates are selected among candidates located in the positions depicted in FIG. 12. The order of derivation is A₁, B₁, B₀, A₀ and B₂. Position B₂ is considered only when any PU of position A₁, B₁, B₀, A₀ is not available (e.g. because it belongs to another slice or tile) or is not coded with IBC mode. After candidate at position A₁ is added, the insertion of the remaining candidates is subject to a redundancy check which ensures that candidates with same motion information are excluded from the list so that coding efficiency is improved. To reduce computational complexity, not all possible candidate pairs are considered in the mentioned redundancy check. Instead only the

pairs linked with an arrow in FIG. 13 are considered and a candidate is only added to the list if the corresponding candidate used for redundancy check has not the same motion information.

[0050] After insertion of the spatial candidates, if the IBC merge list size is still smaller than the maximum IBC merge list size, IBC candidates from HMVP table may be inserted.

Redundancy check are performed when inserting the HMVP candidates.

[0051] Finally, pairwise average candidates are inserted into the IBC merge list.

[0052] When a reference block identified by a merge candidate is outside of the picture, or overlaps with the current block, or outside of the reconstructed area, or outside of the valid area restricted by some constraints, the merge candidate is called **invalid merge candidate**.

[0053] It is noted that invalid merge candidates may be inserted into the IBC merge list.

2.3.2 IBC AMVP mode

[0054] In IBC AMVP mode, an AMVP index point to an entry in the IBC AMVP list is parsed from the bitstream. The construction of the IBC AMVP list can be summarized according to the following sequence of steps:

- Step 1: Derivation of spatial candidates
 - Check A_0 , A_1 until an available candidate is found.
 - Check B_0 , B_1 , B_2 until an available candidate is found.
- Step 2: Insertion of HMVP candidates
- Step 3: Insertion of zero candidates

[0055] After insertion of the spatial candidates, if the IBC AMVP list size is still smaller than the maximum IBC AMVP list size, IBC candidates from HMVP table may be inserted.

[0056] Finally, zero candidates are inserted into the IBC AMVP list.

2.4 Palette Mode

[0057] The basic idea behind a palette mode is that the samples in the CU are represented by a small set of representative colour values. This set is referred to as the palette. It is also possible to indicate a sample that is outside the palette by signalling an escape symbol followed by (possibly quantized) component values. This is illustrated in FIG. 2.

2.5 Palette Mode in HEVC Screen Content Coding extensions (HEVC-SCC)

[0058] In the palette mode in HEVC-SCC, a predictive way is used to code the palette and index map.

2.5.1 Coding of the palette entries

[0059] For coding of the palette entries, a palette predictor is maintained. The maximum size of the palette as well as the palette predictor is signaled in the SPS. In HEVC-SCC, a `palette_predictor_initializer_present_flag` is introduced in the PPS. When this flag is 1, entries for initializing the palette predictor are signalled in the bitstream. The palette predictor is initialized at the beginning of each CTU row, each slice and each tile. Depending on the value of the `palette_predictor_initializer_present_flag`, the palette predictor is reset to 0 or initialized using the palette predictor initializer entries signaled in the PPS. In HEVC-SCC, a palette predictor initializer of size 0 was enabled to allow explicit disabling of the palette predictor initialization at the PPS level.

[0060] For each entry in the palette predictor, a reuse flag is signaled to indicate whether it is part of the current palette. This is illustrated in FIG. 3 The reuse flags are sent using run-length coding of zeros. After this, the number of new palette entries are signaled using exponential Golomb code of order 0. Finally, the component values for the new palette entries are signaled.

2.5.2 Coding of palette indices

[0061] The palette indices are coded using horizontal and vertical traverse scans as shown in FIG. 4. The scan order is explicitly signalled in the bitstream using the `palette_transpose_flag`. For the rest of the subsection it is assumed that the scan is horizontal.

[0062] The palette indices are coded using two main palette sample modes: 'INDEX' and 'COPY_ABOVE'. As explained previously, the escape symbol is also signaled as an 'INDEX' mode and assigned an index equal to the maximum palette size. The mode is signaled using a flag except for the top row or when the previous mode was 'COPY_ABOVE'. In the 'COPY_ABOVE' mode, the palette index of the sample in the row above is copied. In the 'INDEX' mode, the palette index is explicitly signaled. For both 'INDEX' and 'COPY_ABOVE' modes, a run value is signaled which specifies the number of *subsequent* samples that are also coded using the same mode. When escape symbol is part of the run in 'INDEX' or 'COPY_ABOVE' mode, the escape component values are signaled for each escape symbol. The

coding of palette indices is illustrated in FIG. 5.

[0063] This syntax order is accomplished as follows. First, the number of index values for the CU is signaled. This is followed by signaling of the actual index values for the entire CU using truncated binary coding. Both the number of indices as well as the index values are coded in bypass mode. This groups the index-related bypass bins together. Then the palette sample mode (if necessary) and run are signaled in an interleaved manner. Finally, the component escape values corresponding to the escape samples for the entire CU are grouped together and coded in bypass mode.

[0064] An additional syntax element, `last_run_type_flag`, is signaled after signaling the index values. This syntax element, in conjunction with the number of indices, eliminates the need to signal the run value corresponding to the last run in the block.

[0065] In HEVC-SCC, the palette mode is also enabled for 4:2:2, 4:2:0, and monochrome chroma formats. The signaling of the palette entries and palette indices is almost identical for all the chroma formats. In case of non-monochrome formats, each palette entry consists of 3 components. For the monochrome format, each palette entry consists of a single component. For subsampled chroma directions, the chroma samples are associated with luma sample indices that are divisible by 2. After reconstructing the palette indices for the CU, if a sample has only a single component associated with it, only the first component of the palette entry is used. The only difference in signaling is for the escape component values. For each escape sample, the number of escape component values signaled may be different depending on the number of components associated with that sample.

2.6 Coefficients Coding in Transform Skip mode

[0066] In JVET-M0464 and JVET-N0280, several modifications are proposed on the coefficients coding in transform skip (TS) mode in order to adapt the residual coding to the statistics and signal characteristics of the transform skip levels.

[0067] The proposed modifications are listed as follows.

[0068] **No last significant scanning position:** Since the residual signal reflects the spatial residual after the prediction and no energy compaction by transform is performed for TS, the higher probability for trailing zeros or insignificant levels at the bottom right corner of the transform block is not given anymore. Thus, last significant scanning position signalling is

omitted in this case. Instead, the first subblock to be processed is the most bottom right subblock within the transform block

[0069] Subblock CBFs: The absence of the last significant scanning position signalling requires the subblock CBF signalling with `coded_sub_block_flag` for TS to be modified as follows:

- Due to quantization, the aforementioned sequence of insignificance may still occur locally inside a transform block. Thus, the last significant scanning position is removed as described before and **`coded_sub_block_flag` is coded for all sub-blocks.**
- The `coded_sub_block_flag` for the subblock covering the DC frequency position (top-left subblock) presents a special case. In VVC Draft 3, the `coded_sub_block_flag` for this subblock is never signaled and always inferred to be equal to 1. When the last significant scanning position is located in another subblock, it means that there is at least one significant level outside the DC subblock. Consequently, the DC subblock may contain only zero/non-significant levels although the `coded_sub_block_flag` for this subblock is inferred to be equal to 1. With the absence of the last scanning position information in TS, the `coded_sub_block_flag` for each subblock is signaled. This also includes the `coded_sub_block_flag` for the DC subblock except when all other `coded_sub_block_flag` syntax elements are already equal to 0. In this case, the DC `coded_sub_block_flag` is inferred to be equal to 1 (`inferDcSbCbf=1`). Since there has to be at least one significant level in this DC subblock, the **`sig_coeff_flag` syntax element for the first position at (0,0) is not signaled and derived to be equal to 1** (`inferSbDcSigCoeffFlag=1`) instead if all other `sig_coeff_flag` syntax elements in this DC subblock are equal to 0.
- The context modeling for `coded_sub_block_flag` is changed. The **context model index is calculated as the sum of the `coded_sub_block_flag` to the left and the `coded_sub_block_flag` above** the current subblock instead of and a logical disjunction of both.

[0070] `sig_coeff_flag` context modelling: The local template in `sig_coeff_flag` context modeling is modified to only include the neighbor to the left (NB_0) and the neighbor above (NB_1) the current scanning position. The context model offset is just the number of significant

neighboring positions $\text{sig_coeff_flag}[\text{NB}_0] + \text{sig_coeff_flag}[\text{NB}_1]$. Hence, the selection of different context sets depending on the diagonal d within the current transform block is removed. This results in three context models and a single context model set for coding the sig_coeff_flag flag.

[0071] **abs_level_gt1_flag and par_level_flag context modelling:** a single context model is employed for $\text{abs_level_gt1_flag}$ and par_level_flag .

[0072] **abs_remainder coding:** Although the empirical distribution of the transform skip residual absolute levels typically still fits a Laplacian or a Geometrical distribution, there exist larger instationarities than for transform coefficient absolute levels. Particularly, the variance within a window of consecutive realization is higher for the residual absolute levels. This motivates the following modifications of the abs_remainder syntax binarization and context modelling:

- Using a higher cutoff value in the binarization, i.e., the transition point from the coding with sig_coeff_flag , $\text{abs_level_gt1_flag}$, par_level_flag , and $\text{abs_level_gt3_flag}$ to the Rice codes for abs_remainder , and dedicated context models for each bin position yields higher compression efficiency. Increasing the cutoff will result in more "greater than X" flags, e.g. introducing **abs_level_gt5_flag, abs_level_gt7_flag, and so on until a cutoff is reached**. The cutoff itself is fixed to 5 ($\text{numGtFlags}=5$).
- The **template for the rice parameter derivation is modified**, i.e., only the neighbor to the left and the neighbor above the current scanning position are considered similar to the local template for sig_coeff_flag context modeling.

[0073] **coeff_sign_flag context modelling:** Due to the instationarities inside the sequence of signs and the fact that the prediction residual is often biased, the signs can be coded using context models, even when the global empirical distribution is almost uniformly distributed. A single dedicated context model is used for the coding of the signs and the sign is parsed after sig_coeff_flag to keep all context coded bins together.

2.7 Quantized residual Block Differential Pulse-code Modulation(QR-BDPCM)

[0074] In JVET-M0413, a quantized residual block differential pulse-code modulation (QR-

BDPCM) is proposed to code screen contents efficiently.

[0075] The prediction directions used in QR-BDPCM can be vertical and horizontal prediction modes. The intra prediction is done on the entire block by sample copying in prediction direction (horizontal or vertical prediction) similar to intra prediction. The residual is quantized and the delta between the quantized residual and its predictor (horizontal or vertical) quantized value is coded. This can be described by the following: For a block of size M (rows) $\times$ N (cols), let $r_{i,j}$, $0 \leq i \leq M - 1$, $0 \leq j \leq N - 1$ be the prediction residual after performing intra prediction horizontally (copying left neighbor pixel value across the the predicted block line by line) or vertically (copying top neighbor line to each line in the predicted block) using unfiltered samples from above or left block boundary samples. Let $Q(r_{i,j})$, $0 \leq i \leq M - 1$, $0 \leq j \leq N - 1$ denote the quantized version of the residual $r_{i,j}$, where residual is difference between original block and the predicted block values. Then the block DPCM is applied to the quantized residual samples, resulting in modified $M \times N$ array $\tilde{R}$ with elements $\tilde{r}_{i,j}$. When vertical BDPCM is signalled:

$$\tilde{r}_{i,j} = \begin{cases} Q(r_{i,j}), & i = 0, \quad 0 \leq j \leq (N - 1) \\ Q(r_{i,j}) - Q(r_{(i-1),j}), & 1 \leq i \leq (M - 1), \quad 0 \leq j \leq (N - 1) \end{cases} \quad (2-7-1)$$

[0076] For horizontal prediction, similar rules apply, and the residual quantized samples are obtained by

$$\tilde{r}_{i,j} = \begin{cases} Q(r_{i,j}), & 0 \leq i \leq (M - 1), \quad j = 0 \\ Q(r_{i,j}) - Q(r_{i,(j-1)}), & 0 \leq i \leq (M - 1), \quad 1 \leq j \leq (N - 1) \end{cases} \quad (2-7-2)$$

[0077] The residual quantized samples $\tilde{r}_{i,j}$ are sent to the decoder.

[0078] On the decoder side, the above calculations are reversed to produce $Q(r_{i,j})$, $0 \leq i \leq M - 1$, $0 \leq j \leq N - 1$. For vertical prediction case,

$$Q(r_{i,j}) = \sum_{k=0}^i \tilde{r}_{k,j}, \quad 0 \leq i \leq (M - 1), \quad 0 \leq j \leq (N - 1) \quad (2-7-3)$$

For horizontal case,

$$Q(r_{i,j}) = \sum_{k=0}^j \tilde{r}_{i,k}, \quad 0 \leq i \leq (M - 1), \quad 0 \leq j \leq (N - 1) \quad (2-7-4)$$

[0079] The inverse quantized residuals, $Q^{-1}(Q(r_{i,j}))$, are added to the intra block prediction values to produce the reconstructed sample values.

[0080] The main benefit of this scheme is that the inverse DPCM can be done on the fly during coefficient parsing simply adding the predictor as the coefficients are parsed or it can be performed after parsing.

[0081] The draft text changes of QR-BDPCM are shown as follows.

7.3.6.5 Coding unit syntax

	Descriptor
coding_unit (x0, y0, cbWidth, cbHeight, treeType) {	
if(tile_group_type != I sps_ibc_enabled_flag) {	
if(treeType != DUAL_TREE_CHROMA)	
cu_skip_flag [x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] == 0 && tile_group_type != I)	
pred_mode_flag	ae(v)
if(((tile_group_type == I && cu_skip_flag[x0][y0] == 0) (tile_group_type != I && CuPredMode[x0][y0] != MODE_INTRA)) && sps_ibc_enabled_flag)	
pred_mode_ibc_flag	ae(v)
}	
if(CuPredMode[x0][y0] == MODE_INTRA) {	
if(pred_mode_flag == MODE_INTRA && (cIdx == 0) && (cbWidth <= 32) && (CbHeight <= 32)) {	
bdpcm_flag [x0][y0]	ae(v)
if(bdpcm_flag[x0][y0]) {	
bdpcm_dir_flag [x0][y0]	ae(v)
}	
else {	
if(sps_pcm_enabled_flag && cbWidth >= MinIpcmCbSizeY && cbWidth <= MaxIpcmCbSizeY && cbHeight >= MinIpcmCbSizeY && cbHeight <= MaxIpcmCbSizeY)	
pcm_flag [x0][y0]	ae(v)
if(pcm_flag[x0][y0]) {	
while(!byte_aligned())	
pcm_alignment_zero_bit	f(1)

pcm_sample(cbWidth, cbHeight, treeType)	
} else {	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA) {	
if((y0 % CtbSizeY) > 0)	
intra_luma_ref_idx [x0][y0]	ae(v)
if ((intra_luma_ref_idx[x0][y0] == 0 && (cbWidth <= MaxTbSizeY cbHeight <= MaxTbSizeY) && (cbWidth * cbHeight > MinTbSizeY * MinTbSizeY))	
intra_subpartitions_mode_flag [x0][y0]	ae(v)
if(intra_subpartitions_mode_flag[x0][y0] == 1 && cbWidth <= MaxTbSizeY && cbHeight <= MaxTbSizeY)	
intra_subpartitions_split_flag [x0][y0]	ae(v)
if(intra_luma_ref_idx[x0][y0] == 0 && intra_subpartitions_mode_flag[x0][y0] == 0)	
intra_luma_mpm_flag [x0][y0]	ae(v)
if(intra_luma_mpm_flag[x0][y0])	
intra_luma_mpm_idx [x0][y0]	ae(v)
else	
intra_luma_mpm_remainder [x0][y0]	ae(v)
}	
}	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_CHROMA)	
intra_chroma_pred_mode [x0][y0]	ae(v)
}	
} else if(treeType != DUAL_TREE_CHROMA) { /* MODE_INTER or MODE_IBC */	
...	
}	

bdpcm_flag[x0][y0] equal to 1 specifies that a **bdpcm_dir_flag** is present in the coding unit including the luma coding block at the location (x0, y0)

bdpcm_dir_flag[x0][y0] equal to 0 specifies that the prediction direction to be used in a **bdpcm** block is horizontal, otherwise it is vertical.

2.8 Matrix based intra prediction (MIP)

[0082] The matrix based intra prediction is also called affine linear weighted intra prediction (ALWIP), which use a weighted matrix to derive the intra prediction signal.

2.8.1 Description of the method

[0083] For predicting the samples of a rectangular block of width W and height H , affine linear weighted intra prediction (ALWIP) takes one line of H reconstructed neighbouring boundary samples left of the block and one line of W reconstructed neighbouring boundary samples above the block as input. If the reconstructed samples are unavailable, they are generated as it is done in the conventional intra prediction.

The generation of the prediction signal is based on the following three steps:

1. Out of the boundary samples, four samples in the case of $W = H = 4$ and eight samples in all other cases are extracted by averaging.
2. A matrix vector multiplication, followed by addition of an offset, is carried out with the averaged samples as an input. The result is a reduced prediction signal on a subsampled set of samples in the original block.
3. The prediction signal at the remaining positions is generated from the prediction signal on the subsampled set by linear interpolation which is a single step linear interpolation in each direction.

[0084] The matrices and offset vectors needed to generate the prediction signal are taken from three sets S_0, S_1, S_2 of matrices. The set S_0 consists of 18 matrices $A_0^i, i \in \{0, \dots, 17\}$ each of which has 16 rows and 4 columns and 18 offset vectors $b_0^i, i \in \{0, \dots, 17\}$ each of size 16. Matrices and offset vectors of that set are used for blocks of size 4×4 . The set S_1 consists of 10 matrices $A_1^i, i \in \{0, \dots, 9\}$, each of which has 16 rows and 8 columns and 10 offset vectors $b_1^i, i \in \{0, \dots, 9\}$ each of size 16. Matrices and offset vectors of that set are used for blocks of sizes $4 \times 8, 8 \times 4$ and 8×8 . Finally, the set S_2 consists of 6 matrices $A_2^i, i \in \{0, \dots, 5\}$, each of which has 64 rows and 8 columns and of 6 offset vectors $b_2^i, i \in \{0, \dots, 5\}$ of size 64. Matrices and offset vectors of that set or parts of these matrices and offset vectors are used for all other block-shapes.

[0085] The total number of multiplications needed in the computation of the matrix vector

product is always smaller than or equal to $4 \cdot W \cdot H$. In other words, at most four multiplications per sample are required for the ALWIP modes.

2.8.2 Averaging of the boundary

[0086] In a first step, the input boundaries $bdry^{top}$ and $bdry^{left}$ are reduced to smaller boundaries $bdry_{red}^{top}$ and $bdry_{red}^{left}$. Here, $bdry_{red}^{top}$ and $bdry_{red}^{left}$ both consists of 2 samples in the case of a 4×4 -block and both consist of 4 samples in all other cases.

[0087] In the case of a 4×4 -block, for $0 \leq i < 2$, one defines

$$bdry_{red}^{top}[i] = \left(\left(\sum_{j=0}^1 bdry^{top}[i \cdot 2 + j] \right) + 1 \right) \gg 1$$

[0088] and defines $bdry_{red}^{left}$ analogously.

[0089] Otherwise, if the block-width W is given as $W = 4 \cdot 2^k$, for $0 \leq i < 4$, one defines

$$bdry_{red}^{top}[i] = \left(\left(\sum_{j=0}^{2^k-1} bdry^{top}[i \cdot 2^k + j] \right) + (1 \ll (k-1)) \right) \gg k$$

[0090] and defines $bdry_{red}^{left}$ analogously.

[0091] The two reduced boundaries $bdry_{red}^{top}$ and $bdry_{red}^{left}$ are concatenated to a reduced boundary vector $bdry_{red}$ which is thus of size four for blocks of shape 4×4 and of size eight for blocks of all other shapes. If *mode* refers to the ALWIP-mode, this concatenation is defined as follows:

$$bdry_{red} = \begin{cases} [bdry_{red}^{top}, bdry_{red}^{left}] & \text{for } W = H = 4 \text{ and } mode < 18 \\ [bdry_{red}^{left}, bdry_{red}^{top}] & \text{for } W = H = 4 \text{ and } mode \geq 18 \\ [bdry_{red}^{top}, bdry_{red}^{left}] & \text{for } \max(W, H) = 8 \text{ and } mode < 10 \\ [bdry_{red}^{left}, bdry_{red}^{top}] & \text{for } \max(W, H) = 8 \text{ and } mode \geq 10 \\ [bdry_{red}^{top}, bdry_{red}^{left}] & \text{for } \max(W, H) > 8 \text{ and } mode < 6 \\ [bdry_{red}^{left}, bdry_{red}^{top}] & \text{for } \max(W, H) > 8 \text{ and } mode \geq 6. \end{cases}$$

[0092] Finally, for the interpolation of the subsampled prediction signal, on large blocks a second version of the averaged boundary is needed. Namely, if $\min(W, H) > 8$ and $W \geq H$, one writes $W = 8 \cdot 2^l$, and, for $0 \leq i < 8$, defines

$$bdry_{redll}^{top}[i] = \left(\left(\sum_{j=0}^{2^l-1} bdry^{top}[i \cdot 2^l + j] \right) + (1 \ll (l-1)) \right) \gg l.$$

[0093] If $\min(W, H) > 8$ and $H > W$, one defines $bdry_{redll}^{left}$ analogously.

2.8.3 Generation of the reduced prediction signal by matrix vector multiplication

[0094] Out of the reduced input vector $bdry_{red}$ one generates a reduced prediction signal $pred_{red}$. The latter signal is a signal on the downsampled block of width W_{red} and height H_{red} . Here, W_{red} and H_{red} are defined as:

$$W_{red} = \begin{cases} 4 & \text{for } \max(W, H) \leq 8 \\ \min(W, 8) & \text{for } \max(W, H) > 8 \end{cases}$$

$$H_{red} = \begin{cases} 4 & \text{for } \max(W, H) \leq 8 \\ \min(H, 8) & \text{for } \max(W, H) > 8 \end{cases}$$

[0095] The reduced prediction signal $pred_{red}$ is computed by calculating a matrix vector product and adding an offset:

$$pred_{red} = A \cdot bdry_{red} + b.$$

[0096] Here, A is a matrix that has $W_{red} \cdot H_{red}$ rows and 4 columns if $W = H = 4$ and 8 columns in all other cases. b is a vector of size $W_{red} \cdot H_{red}$.

[0097] The matrix A and the vector b are taken from one of the sets S_0, S_1, S_2 as follows. One defines an index $idx = idx(W, H)$ as follows:

$$idx(W, H) = \begin{cases} 0 & \text{for } W = H = 4 \\ 1 & \text{for } \max(W, H) = 8 \\ 2 & \text{for } \max(W, H) > 8. \end{cases}$$

[0098] Moreover, one puts m as follows:

$$m = \begin{cases} mode & \text{for } W = H = 4 \text{ and } mode < 18 \\ mode - 17 & \text{for } W = H = 4 \text{ and } mode \geq 18 \\ mode & \text{for } \max(W, H) = 8 \text{ and } mode < 10 \\ mode - 9 & \text{for } \max(W, H) = 8 \text{ and } mode \geq 10 \\ mode & \text{for } \max(W, H) > 8 \text{ and } mode < 6 \\ mode - 5 & \text{for } \max(W, H) > 8 \text{ and } mode \geq 6. \end{cases}$$

[0099] Then, if $idx \leq 1$ or $idx = 2$ and $\min(W, H) > 4$, one puts $A = A_{idx}^m$ and $b = b_{idx}^m$. In the case that $idx = 2$ and $\min(W, H) = 4$, one lets A be the matrix that arises by leaving out every row of A_{idx}^m that, in the case $W = 4$, corresponds to an odd x-coordinate in the

downsampled block, or, in the case $H = 4$, corresponds to an odd y-coordinate in the downsampled block.

[00100] Finally, the reduced prediction signal is replaced by its transpose in the following cases:

- $W = H = 4$ and $mode \geq 18$
- $\max(W, H) = 8$ and $mode \geq 10$
- $\max(W, H) > 8$ and $mode \geq 6$

[00101] The number of multiplications required for calculation of $pred_{red}$ is 4 in the case of $W = H = 4$ since in this case A has 4 columns and 16 rows. In all other cases, A has 8 columns and $W_{red} \cdot H_{red}$ rows and one immediately verifies that in these cases $8 \cdot W_{red} \cdot H_{red} \leq 4 \cdot W \cdot H$ multiplications are required, i.e. also in these cases, at most 4 multiplications per sample are needed to compute $pred_{red}$.

2.8.4 Illustration of the entire ALWIP process

[00102] The entire process of averaging, matrix vector multiplication and linear interpolation is illustrated for different shapes in FIG. 6, FIG. 7, FIG. 8 and FIG. 9. Note, that the remaining shapes are treated as in one of the depicted cases.

1. Given a 4×4 block, ALWIP takes two averages along each axis of the boundary. The resulting four input samples enter the matrix vector multiplication. The matrices are taken from the set S_0 . After adding an offset, this yields the 16 final prediction samples. Linear interpolation is not necessary for generating the prediction signal. Thus, a total of $(4 \cdot 16)/(4 \cdot 4) = 4$ multiplications per sample are performed.

[00103] FIG. 6 is an illustration of ALWIP for 4×4 blocks.

2. Given an 8×8 block, ALWIP takes four averages along each axis of the boundary. The resulting eight input samples enter the matrix vector multiplication. The matrices are taken from the set S_1 . This yields 16 samples on the odd positions of the prediction block. Thus, a total of $(8 \cdot 16)/(8 \cdot 8) = 2$ multiplications per sample are performed. After adding an offset, these samples are interpolated vertically by using the reduced top boundary. Horizontal interpolation follows by using the original left boundary.

[00104] FIG. 7 is an illustration of ALWIP for 8×8 blocks.

3. Given an 8×4 block, ALWIP takes four averages along the horizontal axis of the boundary and the four original boundary values on the left boundary. The resulting eight input samples enter the matrix vector multiplication. The matrices are taken from the set S_1 . This yields 16 samples on the odd horizontal and each vertical positions of the

prediction block. Thus, a total of $(8 \cdot 16)/(8 \cdot 4) = 4$ multiplications per sample are performed. After adding an offset, these samples are interpolated horizontally by using the original left boundary.

[00105] FIG. 8 is an illustration of ALWIP for 8×4 blocks

[00106] The transposed case is treated accordingly.

4. Given a 16×16 block, ALWIP takes four averages along each axis of the boundary. The resulting eight input samples enter the matrix vector multiplication. The matrices are taken from the set S_2 . This yields 64 samples on the odd positions of the prediction block. Thus, a total of $(8 \cdot 64)/(16 \cdot 16) = 2$ multiplications per sample are performed. After adding an offset, these samples are interpolated vertically by using eight averages of the top boundary. Horizontal interpolation follows by using the original left boundary.

[00107] FIG. 9 is an illustration of ALWIP for 16×16 blocks

For larger shapes, the procedure is essentially the same and it is easy to check that the number of multiplications per sample is less than four.

For $W \times 8$ blocks with $W > 8$, only horizontal interpolation is necessary as the samples are given at the odd horizontal and each vertical position. In this case, $(8 \cdot 64)/(W \cdot 8) = 64/W$ multiplications per sample are performed to calculate the reduced prediction.

Finally for $W \times 4$ blocks with $W > 8$, let A_k be the matrix that arises by leaving out every row that corresponds to an odd entry along the horizontal axis of the downsampled block. Thus, the output size is 32 and again, only horizontal interpolation remains to be performed. For calculation of reduced prediction, $(8 \cdot 32)/(W \cdot 4) = 64/W$ multiplications per sample are performed. For $W = 16$, no additional multiplications are required while, for $W > 16$, less than 2 multiplication per sample are needed for linear interpolation. Thus, total number of multiplications is less than or equal to four.

The transposed cases are treated accordingly.

2.8.5 Single step linear interpolation

[00108] For a $W \times H$ block with $\max(W, H) \geq 8$, the prediction signal arises from the reduced prediction signal $pred_{red}$ on $W_{red} \times H_{red}$ by linear interpolation. Depending on the block shape, linear interpolation is done in vertical, horizontal or both directions. If linear interpolation is to be applied in both directions, it is first applied in horizontal direction if $W < H$ and it is first applied in vertical direction, else.

[00109] Consider without loss of generality a $W \times H$ block with $\max(W, H) \geq 8$ and $W \geq H$. Then, the one-dimensional linear interpolation is performed as follows. Without loss of generality, it suffices to describe linear interpolation in vertical direction. First, the reduced prediction signal is extended to the top by the boundary signal. Define the vertical upsampling

factor $U_{ver} = H/H_{red}$ and write $U_{ver} = 2^{u_{ver}} > 1$. Then, define the extended reduced prediction signal by

$$pred_{red}[x][-1] = \begin{cases} bdry_{red}^{top}[x] & \text{for } W = 8 \\ bdry_{redll}^{top}[x] & \text{for } W > 8. \end{cases}$$

[00110] Then, from this extended reduced prediction signal, the vertically linear interpolated prediction signal is generated by

$$\begin{aligned} pred_{red}^{ups,ver}[x][U_{ver} \cdot y + k] \\ = \left((U_{ver} - k - 1) \cdot pred_{red}[x][y - 1] + (k + 1) \cdot pred_{red}[x][y] + \frac{U_{ver}}{2} \right) \\ \gg u_{ver} \end{aligned}$$

for $0 \leq x < W_{red}$, $0 \leq y < H_{red}$ and $0 \leq k < U_{ver}$.

bit-shift-only linear interpolation algorithm does not require any multiplication.

2.8.6 Signalization of the proposed intra prediction modes

[00111] For each Coding Unit (CU) in intra mode, a flag indicating if an ALWIP mode is to be applied on the corresponding Prediction Unit (PU) or not is sent in the bitstream. If an ALWIP mode is to be applied, the index $predmode$ of the ALWIP mode is signaled using a MPM-list with 3 MPMS.

[00112] Here, the derivation of the MPMS is performed using the intra-modes of the above and the left PU as follows. There are three fixed tables $map_angular_to_alwip_{idx}$, $idx \in \{0,1,2\}$ that assign to each conventional intra prediction mode $predmode_{Angular}$ an ALWIP mode

$$predmode_{ALWIP} = map_angular_to_alwip_{idx}[predmode_{Angular}].$$

[00113] For each PU of width W and height H one defines an index

$$idx(PU) = idx(W, H) \in \{0,1,2\}$$

[00114] that indicates from which of the three sets the ALWIP-parameters are to be taken as in Section 1.3 above.

[00115] If the above Prediction Unit PU_{above} is available, belongs to the same CTU as the current PU and is in intra mode, if $idx(PU) = idx(PU_{above})$ and if ALWIP is applied on PU_{above} with ALWIP-mode $predmode_{ALWIP}^{above}$, one puts

$$mode_{ALWIP}^{above} = predmode_{ALWIP}^{above}.$$

[00116] If the above PU is available, belongs to the same CTU as the current PU and is in intra

mode and if a conventional intra prediction mode $predmode_{Angular}^{above}$ is applied on the above PU, one puts

$$mode_{ALWIP}^{above} = map_angular_to_alwip_{idx(PU_{above})}[predmode_{Angular}^{above}].$$

[00117] In all other cases, one puts

$$mode_{ALWIP}^{above} = -1$$

[00118] which means that this mode is unavailable. In the same way but without the restriction that the left PU needs to belong to the same CTU as the current PU, one derives a mode $mode_{ALWIP}^{left}$.

[00119] Finally, three fixed default lists $list_{idx}$, $idx \in \{0,1,2\}$ are provided, each of which contains three distinct ALWIP modes. Out of the default list $list_{idx(PU)}$ and the modes $mode_{ALWIP}^{above}$ and $mode_{ALWIP}^{left}$, one constructs three distinct MPMs by substituting -1 by default values as well as eliminating repetitions.

2.8.7 Adapted MPM-list derivation for conventional luma and chroma intra-prediction modes

[00120] The proposed ALWIP-modes are harmonized with the MPM-based coding of the conventional intra-prediction modes as follows. The luma and chroma MPM-list derivation processes for the conventional intra-prediction modes uses fixed tables $map_alwip_to_angular_{idx}$, $idx \in \{0,1,2\}$, mapping an ALWIP-mode $predmode_{ALWIP}$ on a given PU to one of the conventional intra-prediction modes

$$predmode_{Angular} = map_alwip_to_angular_{idx(PU)}[predmode_{ALWIP}].$$

[00121] For the luma MPM-list derivation, whenever a neighboring luma block is encountered which uses an ALWIP-mode $predmode_{ALWIP}$, this block is treated as if it was using the conventional intra-prediction mode $predmode_{Angular}$. For the chroma MPM-list derivation, whenever the current luma block uses an LWIP-mode, the same mapping is used to translate the ALWIP-mode to a conventional intra prediction mode.

2.9 Chroma intra mode coding

[00122] For chroma intra mode coding, a total of 8 intra modes are allowed for chroma intra mode coding. Those modes include five traditional intra modes and three cross-component linear

model modes. Chroma DM mode use the corresponding luma intra prediction mode. Since separate block partitioning structure for luma and chroma components is enabled in I slices, one chroma block may correspond to multiple luma blocks. Therefore, for Chroma DM mode, the intra prediction mode of the corresponding luma block covering the center position of the current chroma block is directly inherited.

3. Examples of technical problems solved by disclosed embodiments

[00123] Although the QR-BDPCM can achieve coding benefits on screen content coding, it may still have some drawbacks.

1. The prediction in the QR-BDPCM mode is only limited to horizontal and vertical intra predictions, which may limit the prediction efficiency in the QR-BDPCM mode.
2. The intra prediction mode is signaled for QR-BDPCM coded blocks which may increase the rate cost of the QR-BDPCM mode
3. The neighboring information is not considered when mapping the signaled message to the prediction modes in the QR-BDPCM mode
4. The QR-BDPCM represents the residue by only supporting horizontal DPCM and vertical DPCM, which may comprise the coding performance on complex residual block
5. The residual range in the QR-BDPCM may exceed the maximal range of other non QR-BDPCM modes
6. The QR-BDPCM does not consider the block shape
7. How to handle chroma when the luma block is coded with QR-BDPCM is unknown.

4. Example embodiments and techniques

[00124] The listing of items below should be considered as examples to explain general concepts. These technical features should not be interpreted in a narrow way. Furthermore, these technical features can be combined in any manner.

1. Sample prediction in the QR-BDPCM coded blocks may be generated by the matrix based intra prediction (MIP) method.
 - a. In one example, when QR-BDPCM and MIP are both enabled for one block, it is restricted that only partial of allowed modes in MIP are supported.

- i. In one example, the partial of allowed modes may include those modes associated with the matrix based intra prediction method that could be mapped to horizontal and/or vertical normal intra mode.
 - ii. In one example, the partial of allowed modes may only include those modes associated with the matrix based intra prediction method that could be mapped to horizontal and/or vertical normal intra mode.
 - b. In one example, when QR-BDPCM and MIP are both enabled for one block, all of allowed modes in MIP are supported.
- 2. Sample prediction in the QR-BDPCM coded blocks may be generated by intra prediction modes other than vertical/horizontal intra predictions.
 - a. In one example, the samples in the QR-BDPCM coded blocks may be predicted by an intra prediction mode K
 - i. In one example, K may be PLANAR mode
 - ii. In one example, K may be DC mode
 - iii. In one example, K may be HORIZONTAL mode
 - iv. In one example, K may be VERTICAL mode
 - v. In one example, K may be one candidate in the list of most probable modes.
 - vi. In one example, K may be signaled in the bitstream
 - b. The allowed intra prediction modes for QR-BDPCM may be based on
 - i. A message signalled in the SPS/VPS/PPS/picture header/slice header/tile group header/LCU row/group of LCUs
 - ii. Block dimension of current block and/or its neighboring blocks
 - iii. Block shape of current block and/or its neighboring blocks
 - iv. Prediction modes (Intra/Inter) of the neighboring blocks of the current block
 - v. Intra prediction modes of the neighboring blocks of the current block
 - vi. The indication of QR-BDPCM modes of the neighboring block of the current block
 - vii. Current quantization parameter of current block and/or that of its neighboring blocks
 - viii. Indication of the color format (such as 4:2:0, 4:4:4)

- ix. Separate/dual coding tree structure
 - x. Slice/tile group type and/or picture type
3. Sample prediction in the QR-BDPCM coded blocks may be generated by non-adjacent samples.
- a. In one example, for the IBC merge mode, QR-BDPCM may be also enabled.
 - b. In one example, for the IBC AMVP mode, QR-BDPCM may be also enabled.
 - c. The block vector used in the IBC and QR-BDPCM may be signaled or derived or pre-defined.
 - i. In one example, the IBC mode may be indicated by a motion vector (block vector) and/or a merge index.
 - ii. In one example, the IBC mode may be indicated by a default motion vector.
 - 1. In one example, the default motion vector may be $(-w, 0)$, where w is a positive integer number
 - 2. In one example, the default motion vector may be $(0, -h)$, where h is a positive integer number
 - 3. In one example, the default motion vector may be $(-w, -h)$, where w and h are two positive integer numbers
 - iii. In one example, the indication of a motion vector used in the IBC and QR-BDPCM coded blocks may be based on:
 - 1. A message signalled in the SPS/VPS/PPS/picture header/slice header/tile group header/LCU row/group of LCUs
 - 2. Block dimension of current block and/or its neighboring blocks
 - 3. Block shape of current block and/or its neighboring blocks
 - 4. Prediction modes (Intra/Inter) of the neighboring blocks of the current block
 - 5. Motion vectors of the neighboring blocks of the current block
 - 6. The indication of QR-BDPCM modes of the neighboring block of the current block
 - 7. Current quantization parameter of current block and/or that of its neighboring blocks

8. Indication of the color format (such as 4:2:0, 4:4:4)
 9. Separate/dual coding tree structure
 10. Slice/tile group type and/or picture type
- d. In one example, the sample prediction in the QR-BDPCM mode may be generated by Inter prediction tools (e.g. affine mode, merge mode and inter mode)
4. The indication of the quantized residual prediction direction in the QR-BDPCM may be derived on-the-fly.
 - a. In one example, the indication of the quantized residual prediction direction in the QR-BDPCM may be inferred based on the indication of current intra prediction mode
 - i. In one example, the direction of quantized residual prediction in the QR-BDPCM may be inferred to vertical when the intra prediction mode is vertical
 - ii. In one example, the direction of quantized residual prediction in the QR-BDPCM may be inferred to horizontal when the intra prediction mode is horizontal
 - iii. In one example, the direction of quantized residual prediction in the QR-BDPCM may be inferred to vertical when the intra prediction mode is horizontal
 - iv. In one example, the direction of quantized residual prediction in the QR-BDPCM may be inferred to horizontal when the intra prediction mode is vertical
 - b. In one example, the indication of the quantized residual prediction direction in the QR-BDPCM may be based on
 - i. A message signalled in the SPS/VPS/PPS/picture header/slice header/tile group header/LCU row/group of LCUs
 - ii. Block dimension of current block and/or its neighboring blocks
 - iii. Block shape of current block and/or its neighboring blocks
 - iv. The most probable modes of the current block and/or its neighboring blocks
 - v. Prediction modes (Intra/Inter) of the neighboring blocks of the current block

- vi. Intra prediction modes of the neighboring blocks of the current block
 - vii. Motion vectors of the neighboring blocks of the current block
 - viii. The indication of QR-BDPCM modes of the neighboring block of the current block
 - ix. Current quantization parameter of current block and/or that of its neighboring blocks
 - x. Indication of the color format (such as 4:2:0, 4:4:4)
 - xi. Separate/dual coding tree structure
 - xii. Transform type applied to the current block
 - xiii. Slice/tile group type and/or picture type
5. The mapping from a signaled index in the QR-BDPCM to the intra prediction mode in the QR-BDPCM mode may be based on
- a. A message signalled in the SPS/VPS/PPS/picture header/slice header/tile group header/LCU row/group of LCUs
 - b. Block dimension of current block and/or its neighboring blocks
 - c. Block shape of current block and/or its neighboring blocks
 - d. The most probable modes of the current block and/or its neighboring blocks
 - e. Prediction modes (Intra/Inter) of the neighboring blocks of the current block
 - f. Intra prediction modes of the neighboring blocks of the current block
 - g. Motion vectors of the neighboring blocks of the current block
 - h. The indication of QR-BDPCM modes of the neighboring block of the current block
 - i. Current quantization parameter of current block and/or that of its neighboring blocks
 - j. Indication of the color format (such as 4:2:0, 4:4:4)
 - k. Separate/dual coding tree structure
 - l. Transform type applied to the current block
 - m. Slice/tile group type and/or picture type
6. In QR-BDPCM, the quantized residue are predicted along the horizontal and vertical directions. It is proposed to predict the quantized residue along the directions other than

vertical and horizontal directions. Suppose $Q(r_{i,j})$ denotes the quantized residue $\tilde{r}_{i,j}$ denotes the quantized residue after the residue prediction process.

- a. In one example, the 45-degree QR-BDPCM may be supported.
 - i. In one example, the DPCM may be performed along the 45-degree direction, where the $\tilde{r}_{i,j}$ may be derived by $Q(r_{i,j}) - Q(r_{(i-1),(j-1)})$ if the $Q(r_{(i-1),(j-1)})$ is available.
- b. In one example, the 135-degree QR-BDPCM may be supported.
 - i. In one example, the DPCM may be performed along the 135-degree direction, where the $\tilde{r}_{i,j}$ may be derived by $Q(r_{i,j}) - Q(r_{(i-1),(j+1)})$ if the $Q(r_{(i-1),(j+1)})$ is available
- c. In one example, any directions may be supported in QR-BDPCM.
 - i. In one example, the $\tilde{r}_{i,j}$ may be derived by $Q(r_{i,j}) - Q(r_{(i-m),(j-n)})$ if the $Q(r_{(i-m),(j-n)})$ is available.
 1. In one example, m and/or n may be signalled in the bitstream
 2. In one example, m and/or n may be integer numbers and may be based on
 3. A message signalled in the SPS/VPS/PPS/picture header/slice header/tile group header/LCU row/group of LCUs
 4. i and/or j
 5. Block dimension of current block and/or its neighboring blocks
 6. Block shape of current block and/or its neighboring blocks
 7. The most probable modes of the current block and/or its neighboring blocks
 8. Prediction modes (Intra/Inter) of the neighboring blocks of the current block
 9. Intra prediction modes of the neighboring blocks of the current block
 10. Motion vectors of the neighboring blocks of the current block
 11. The indication of QR-BDPCM modes of the neighboring block of the current block

12. Current quantization parameter of current block and/or that of its neighboring blocks
 13. Indication of the color format (such as 4:2:0, 4:4:4)
 14. Separate/dual coding tree structure
 15. Slice/tile group type and/or picture type
7. QR-BDPCM may be applied to chroma blocks (e.g., Cb/Cr, or B/R color components).
- a. In one example, the allowed intra prediction directions for luma and chroma QR-BDPCM coded blocks may be the same, e.g., only horizontal and vertical.
 - b. In one example, the allowed prediction methods for luma and chroma QR-BDPCM coded blocks may be the same, e.g., IBC/Inter/ horizontal and vertical intra prediction modes.
 - c. In one example, the allowed residual prediction direction for luma and chroma QR-BDPCM coded blocks may be the same.
 - d. In one example, the residual prediction direction for chroma QR-BDPCM may be derived from the residual prediction direction for corresponding luma block.
 - i. In one example, the corresponding luma block may be the collocated luma block.
 - ii. In one example, the corresponding luma block may be the luma block containing the collocated sample of the upper-left corner of the chroma block.
 - iii. In one example, the corresponding luma block may be the luma block containing the collocated sample of the centered sample of the chroma block.
 - e. In one example, CCLM and QR-BDPCM couldn't be applied to the same chroma block.
 - i. Alternatively, CCLM may be also applicable to QR-BDPCM coded blocks.
 - f. In one example, joint chroma residual coding (e.g., joint cb and cr coding) method and QR-BDPCM couldn't be applied to the same chroma block.
8. The reconstructed quantized residue in QR-BDPCM may be restricted to be within a specific range.

- a. In one example, a constraint may be added that all the quantized residual differences (e.g., $\tilde{r}_{i,j}$ in equation 2-7-1 and 2-7-2) may be within a specific range.
- b. In one example, a constraint may be added that all the reconstructed quantized residual (e.g., $Q(r_{i,j})$ in equation 2-7-3 and 2-7-4) may be within a specific range.
- c. In one example, clipping operation may be applied to the quantized residual differences (e.g., $\tilde{r}_{i,j}$ in equation 2-7-1 and 2-7-2) so that the reconstructed quantized residual may be within a specific range.
- d. In one example, clipping operation may be applied to the reconstructed quantized residual differences (e.g., $Q(r_{i,j})$ in equation 2-7-3 and 2-7-4) so that the reconstructed quantized residual may be within a specific range.
- e. In one example, the clipping operation may be defined as $(x < \min? \min: (x > \max? \max: x))$
- f. In one example, the clipping operation may be defined as $(x \leq \min? \min: (x \geq \max? \max: x))$
- g. In one example, the clipping operation may be defined as $(x < \min? \min: (x \geq \max? \max: x))$
- h. In one example, the clipping operation may be defined as $(x \leq \min? \min: (x > \max? \max: x))$
- i. In one example, the min and/or max may be negative or positive
- j. In one example, min is set to -32768 and max is set to 32767.
 - i. Alternatively, the min and/or max may depend on the range of inverse quantization for blocks coded not with QR-BDPCM.
 - ii. Alternatively, the min and/or max may depend on the bit depth of input sample/reconstructed sample.
 - iii. Alternatively, the min and/or max may depend on whether lossless coding is used.
 1. In one example, the min and/or max may depend on `transquant_bypass_enabled_flag`.
 2. In one example, the min and/or max may depend on `cu_transquant_bypass_flag`.

- k. In one example, the min and/or max may be based on
 - i. A message signalled in the SPS/VPS/PPS/picture header/slice header/tile group header/LCU row/group of LCUs
 - ii. Block dimension of current block and/or its neighboring blocks
 - iii. Block shape of current block and/or its neighboring blocks
 - iv. The most probable modes of the current block and/or its neighboring blocks
 - v. Prediction modes (Intra/Inter) of the neighboring blocks of the current block
 - vi. Intra prediction modes of the neighboring blocks of the current block
 - vii. Motion vectors of the neighboring blocks of the current block
 - viii. The indication of QR-BDPCM modes of the neighboring block of the current block
 - ix. Current quantization parameter of current block and/or that of its neighboring blocks
 - x. Indication of the color format (such as 4:2:0, 4:4:4)
 - xi. Separate/dual coding tree structure
 - xii. Transform type applied to the current block
 - xiii. Slice/tile group type and/or picture type
- 9. QR-DPCM may be applied from the last row/column to the first row/column for a block.
 - a. In one example, when the residual prediction direction is horizontal, the (i+1)-th column's residual may be used to predict the i-th column's residual.
 - b. In one example, when the residual prediction direction is vertical, the (i+1)-th row's residual may be used to predict the i-th row's residual.
- 10. QR-DPCM may be applied to a subset of a block.
 - a. In one example, when the residual prediction direction is horizontal, QR-DPCM does not apply to the left most k columns of residues.
 - b. In one example, when the residual prediction direction is vertical, QR-DPCM does not apply to the upper most k rows of residues.
 - c. In one example, when the residual prediction direction is horizontal, QR-DPCM does not apply to the right most k columns of residues.

- d. In one example, when the residual prediction direction is vertical, QR-DPCM does not apply to the bottom most k rows of residues.
- e. The value of k described above may be a predefined value, based on.
 - i. A message signalled in the SPS/VPS/PPS/picture header/slice header/tile group header/LCU row/group of LCUs
 - ii. Block dimension of current block and/or its neighboring blocks
 - iii. Block shape of current block and/or its neighboring blocks
 - iv. The most probable modes of the current block and/or its neighboring blocks
 - v. Prediction modes (Intra/Inter) of the neighboring blocks of the current block
 - vi. Intra prediction modes of the current block
 - vii. Intra prediction modes of the neighboring blocks of the current block
 - viii. Motion vectors of the neighboring blocks of the current block
 - ix. The indication of QR-BDPCM modes of the neighboring block of the current block
 - x. Current quantization parameter of current block and/or that of its neighboring blocks
 - xi. Indication of the color format (such as 4:2:0, 4:4:4)
 - xii. Separate/dual coding tree structure
 - xiii. Transform type applied to the current block
 - xiv. Slice/tile group type and/or picture type

11. QR-DPCM may be applied segment by segment for a block

- a. In one example, when the residual prediction direction is vertical and $N=nK$, residual prediction may be performed as

$$\tilde{r}_{i,j} = \begin{cases} Q(r_{i,j}), & i \% K = 0, \quad 0 \leq j \leq (N - 1) \\ Q(r_{i,j}) - Q(r_{(i-1),j}), & i \% K > 0, \quad i \leq (M - 1), \quad 0 \leq j \leq (N - 1) \end{cases}$$

- b. In one example, when the residual prediction direction is horizontal and $M=mK$, residual prediction may be performed as

$$\tilde{r}_{i,j} = \begin{cases} Q(r_{i,j}), & 0 \leq i \leq (M - 1), \quad j \% K = 0 \\ Q(r_{i,j}) - Q(r_{i,(j-1)}), & 0 \leq i \leq (M - 1), \quad j \% K > 0, \quad j \leq (N - 1) \end{cases}$$

12. Enabling/disabling QR-DPCM for one color component may be derived from that associated with another color component.
- a. In one example, for a chroma block, whether to enable QR-DPCM may be dependent on the usage of QR-DPCM associated with one or multiple representative blocks within the collocated luma block.
 - i. In one example, the representative block may be defined in the same way as that used for DM derivation.
 - ii. In one example, if the representative block within the collocated luma block is QR-DPCM coded, and the current chroma block is coded with DM mode, QR-DPCM may be also enabled for the current chroma block.
 - b. Alternatively, indication of usage of QR-DPCM may be signaled for chroma components.
 - i. In one example, one flag may be signaled to indicate the usage for two chroma components.
 - ii. Alternatively, two flags may be signaled to indicate the usage for two chroma components, respectively.
 - iii. In one example, when the chroma block is coded with certain modes (such as CCLM), the signaling of indication of usage of QR-DPCM is skipped.
13. The above methods may be also applicable to other variances of DPCM/QR-DPCM.

[00125] 5. Reference

- [1] ITU-T and ISO/IEC, “High efficiency video coding”, Rec. ITU-T H.265 | ISO/IEC 23008-2 (02/2018).
- [2] B. Bross, J. Chen, S. Liu, Versatile Video Coding (Draft 4), JVET-M1001, Jan.2019

[00126] FIG. 1000 is a block diagram of a video processing apparatus 1000. The apparatus 1000 may be used to implement one or more of the methods described herein. The apparatus 1000 may be embodied in a smartphone, tablet, computer, Internet of Things (IoT) receiver, and so on. The apparatus 1000 may include one or more processors 1002, one or more memories 1004 and video processing hardware 1006. The processor(s) 1002 may be configured to implement one or more methods described in the present document. The memory (memories) 1004 may be used

for storing data and code used for implementing the methods and techniques described herein. The video processing hardware 1006 may be used to implement, in hardware circuitry, some techniques described in the present document.

[00127] FIG. 11 is a flowchart for an example method 1100 of video processing. The method 1100 includes performing (1102) a conversion between a current video block and a bitstream representation of the current video block using a differential coding mode and selectively using an intra prediction mode based on a coexistence rule. The intra prediction mode is used for generating predictions for samples of the current video block. The differential coding mode is used to represent a quantized residual block from the predictions of the pixels, using a differential pulse coding modulation representation.

[00128] FIG. 14 is a block diagram showing an example video processing system 1400 in which various techniques disclosed herein may be implemented. Various implementations may include some or all of the components of the system 1400. The system 1400 may include input 1402 for receiving video content. The video content may be received in a raw or uncompressed format, e.g., 8 or 10 bit multi-component pixel values, or may be in a compressed or encoded format. The input 1402 may represent a network interface, a peripheral bus interface, or a storage interface. Examples of network interface include wired interfaces such as Ethernet, passive optical network (PON), etc. and wireless interfaces such as Wi-Fi or cellular interfaces.

[00129] The system 1400 may include a coding component 1404 that may implement the various coding or encoding methods described in the present document. The coding component 1404 may reduce the average bitrate of video from the input 1402 to the output of the coding component 1404 to produce a coded representation of the video. The coding techniques are therefore sometimes called video compression or video transcoding techniques. The output of the coding component 1404 may be either stored, or transmitted via a communication connected, as represented by the component 1406. The stored or communicated bitstream (or coded) representation of the video received at the input 1402 may be used by the component 1408 for generating pixel values or displayable video that is sent to a display interface 1410. The process of generating user-viewable video from the bitstream representation is sometimes called video decompression. Furthermore, while certain video processing operations are referred to as “coding” operations or tools, it will be appreciated that the coding tools or operations are used at

an encoder and corresponding decoding tools or operations that reverse the results of the coding will be performed by a decoder.

[00130] Examples of a peripheral bus interface or a display interface may include universal serial bus (USB) or high definition multimedia interface (HDMI) or Displayport, and so on. Examples of storage interfaces include SATA (serial advanced technology attachment), PCI, IDE interface, and the like. The techniques described in the present document may be embodied in various electronic devices such as mobile phones, laptops, smartphones or other devices that are capable of performing digital data processing and/or video display.

[00131] Some embodiments of the disclosed technology include making a decision or determination to enable a video processing tool or mode. In an example, when the video processing tool or mode is enabled, the encoder will use or implement the tool or mode in the processing of a block of video, but may not necessarily modify the resulting bitstream based on the usage of the tool or mode. That is, a conversion from the block of video to the bitstream representation of the video will use the video processing tool or mode when it is enabled based on the decision or determination. In another example, when the video processing tool or mode is enabled, the decoder will process the bitstream with the knowledge that the bitstream has been modified based on the video processing tool or mode. That is, a conversion from the bitstream representation of the video to the block of video will be performed using the video processing tool or mode that was enabled based on the decision or determination.

[00132] Some embodiments of the disclosed technology include making a decision or determination to disable a video processing tool or mode. In an example, when the video processing tool or mode is disabled, the encoder will not use the tool or mode in the conversion of the block of video to the bitstream representation of the video. In another example, when the video processing tool or mode is disabled, the decoder will process the bitstream with the knowledge that the bitstream has not been modified using the video processing tool or mode that was disabled based on the decision or determination.

[00133] FIG. 15 is a block diagram that illustrates an example video coding system 100 that may utilize the techniques of this disclosure. As shown in FIG. 15, video coding system 100 may include a source device 110 and a destination device 120. Source device 110 generates encoded video data which may be referred to as a video encoding device. Destination device 120 may

decode the encoded video data generated by source device 110 which may be referred to as a video decoding device. Source device 110 may include a video source 112, a video encoder 114, and an input/output (I/O) interface 116.

[00134] Video source 112 may include a source such as a video capture device, an interface to receive video data from a video content provider, and/or a computer graphics system for generating video data, or a combination of such sources. The video data may comprise one or more pictures. Video encoder 114 encodes the video data from video source 112 to generate a bitstream. The bitstream may include a sequence of bits that form a coded representation of the video data. The bitstream may include coded pictures and associated data. The coded picture is a coded representation of a picture. The associated data may include sequence parameter sets, picture parameter sets, and other syntax structures. I/O interface 116 may include a modulator/demodulator (modem) and/or a transmitter. The encoded video data may be transmitted directly to destination device 120 via I/O interface 116 through network 130a. The encoded video data may also be stored onto a storage medium/server 130b for access by destination device 120.

[00135] Destination device 120 may include an I/O interface 126, a video decoder 124, and a display device 122.

[00136] I/O interface 126 may include a receiver and/or a modem. I/O interface 126 may acquire encoded video data from the source device 110 or the storage medium/ server 130b. Video decoder 124 may decode the encoded video data. Display device 122 may display the decoded video data to a user. Display device 122 may be integrated with the destination device 120, or may be external to destination device 120 which be configured to interface with an external display device.

[00137] Video encoder 114 and video decoder 124 may operate according to a video compression standard, such as the High Efficiency Video Coding (HEVC) standard, Versatile Video Coding(VVM) standard and other current and/or further standards.

[00138] FIG. 16 is a block diagram illustrating an example of video encoder 200, which may be video encoder 114 in the system 100 illustrated in FIG. 15.

[00139] Video encoder 200 may be configured to perform any or all of the techniques of this disclosure. In the example of FIG. 16, video encoder 200 includes a plurality of functional

components. The techniques described in this disclosure may be shared among the various components of video encoder 200. In some examples, a processor may be configured to perform any or all of the techniques described in this disclosure.

[00140] The functional components of video encoder 200 may include a partition unit 201, a predication unit 202 which may include a mode select unit 203, a motion estimation unit 204, a motion compensation unit 205 and an intra prediction unit 206, a residual generation unit 207, a transform unit 208, a quantization unit 209, an inverse quantization unit 210, an inverse transform unit 211, a reconstruction unit 212, a buffer 213, and an entropy encoding unit 214.

[00141] In other examples, video encoder 200 may include more, fewer, or different functional components. In an example, predication unit 202 may include an intra block copy(IBC) unit. The IBC unit may perform predication in an IBC mode in which at least one reference picture is a picture where the current video block is located.

[00142] Furthermore, some components, such as motion estimation unit 204 and motion compensation unit 205 may be highly integrated, but are represented in the example of FIG. 16 separately for purposes of explanation.

[00143] Partition unit 201 may partition a picture into one or more video blocks. Video encoder 200 and video decoder 300 may support various video block sizes.

[00144] Mode select unit 203 may select one of the coding modes, intra or inter, e.g., based on error results, and provide the resulting intra- or inter-coded block to a residual generation unit 207 to generate residual block data and to a reconstruction unit 212 to reconstruct the encoded block for use as a reference picture. In some example, Mode select unit 203 may select a combination of intra and inter predication (CIIP) mode in which the predication is based on an inter predication signal and an intra predication signal. Mode select unit 203 may also select a resolution for a motion vector (e.g., a sub-pixel or integer pixel precision) for the block in the case of inter-predication.

[00145] To perform inter prediction on a current video block, motion estimation unit 204 may generate motion information for the current video block by comparing one or more reference frames from buffer 213 to the current video block. Motion compensation unit 205 may determine a predicted video block for the current video block based on the motion information and decoded samples of pictures from buffer 213 other than the picture associated with the current video

block.

[00146] Motion estimation unit 204 and motion compensation unit 205 may perform different operations for a current video block, for example, depending on whether the current video block is in an I slice, a P slice, or a B slice.

[00147] In some examples, motion estimation unit 204 may perform uni-directional prediction for the current video block, and motion estimation unit 204 may search reference pictures of list 0 or list 1 for a reference video block for the current video block. Motion estimation unit 204 may then generate a reference index that indicates the reference picture in list 0 or list 1 that contains the reference video block and a motion vector that indicates a spatial displacement between the current video block and the reference video block. Motion estimation unit 204 may output the reference index, a prediction direction indicator, and the motion vector as the motion information of the current video block. Motion compensation unit 205 may generate the predicted video block of the current block based on the reference video block indicated by the motion information of the current video block.

[00148] In other examples, motion estimation unit 204 may perform bi-directional prediction for the current video block, motion estimation unit 204 may search the reference pictures in list 0 for a reference video block for the current video block and may also search the reference pictures in list 1 for another reference video block for the current video block. Motion estimation unit 204 may then generate reference indexes that indicate the reference pictures in list 0 and list 1 containing the reference video blocks and motion vectors that indicate spatial displacements between the reference video blocks and the current video block. Motion estimation unit 204 may output the reference indexes and the motion vectors of the current video block as the motion information of the current video block. Motion compensation unit 205 may generate the predicted video block of the current video block based on the reference video blocks indicated by the motion information of the current video block.

[00149] In some examples, motion estimation unit 204 may output a full set of motion information for decoding processing of a decoder.

[00150] In some examples, motion estimation unit 204 may do not output a full set of motion information for the current video. Rather, motion estimation unit 204 may signal the motion information of the current video block with reference to the motion information of another video

block. For example, motion estimation unit 204 may determine that the motion information of the current video block is sufficiently similar to the motion information of a neighboring video block.

[00151] In one example, motion estimation unit 204 may indicate, in a syntax structure associated with the current video block, a value that indicates to the video decoder 300 that the current video block has the same motion information as the another video block.

[00152] In another example, motion estimation unit 204 may identify, in a syntax structure associated with the current video block, another video block and a motion vector difference (MVD). The motion vector difference indicates a difference between the motion vector of the current video block and the motion vector of the indicated video block. The video decoder 300 may use the motion vector of the indicated video block and the motion vector difference to determine the motion vector of the current video block.

[00153] As discussed above, video encoder 200 may predictively signal the motion vector. Two examples of predictive signaling techniques that may be implemented by video encoder 200 include advanced motion vector predication (AMVP) and merge mode signaling.

[00154] Intra prediction unit 206 may perform intra prediction on the current video block. When intra prediction unit 206 performs intra prediction on the current video block, intra prediction unit 206 may generate prediction data for the current video block based on decoded samples of other video blocks in the same picture. The prediction data for the current video block may include a predicted video block and various syntax elements.

[00155] Residual generation unit 207 may generate residual data for the current video block by subtracting (e.g., indicated by the minus sign) the predicted video block(s) of the current video block from the current video block. The residual data of the current video block may include residual video blocks that correspond to different sample components of the samples in the current video block.

[00156] In other examples, there may be no residual data for the current video block for the current video block, for example in a skip mode, and residual generation unit 207 may not perform the subtracting operation.

[00157] Transform processing unit 208 may generate one or more transform coefficient video blocks for the current video block by applying one or more transforms to a residual video block

associated with the current video block.

[00158] After transform processing unit 208 generates a transform coefficient video block associated with the current video block, quantization unit 209 may quantize the transform coefficient video block associated with the current video block based on one or more quantization parameter (QP) values associated with the current video block.

[00159] Inverse quantization unit 210 and inverse transform unit 211 may apply inverse quantization and inverse transforms to the transform coefficient video block, respectively, to reconstruct a residual video block from the transform coefficient video block. Reconstruction unit 212 may add the reconstructed residual video block to corresponding samples from one or more predicted video blocks generated by the predication unit 202 to produce a reconstructed video block associated with the current block for storage in the buffer 213.

[00160] After reconstruction unit 212 reconstructs the video block, loop filtering operation may be performed reduce video blocking artifacts in the video block.

[00161] Entropy encoding unit 214 may receive data from other functional components of the video encoder 200. When entropy encoding unit 214 receives the data, entropy encoding unit 214 may perform one or more entropy encoding operations to generate entropy encoded data and output a bitstream that includes the entropy encoded data.

[00162] FIG. 17 is a block diagram illustrating an example of video decoder 300 which may be video decoder 114 in the system 100 illustrated in FIG. 15.

[00163] The video decoder 300 may be configured to perform any or all of the techniques of this disclosure. In the example of FIG. 17, the video decoder 300 includes a plurality of functional components. The techniques described in this disclosure may be shared among the various components of the video decoder 300. In some examples, a processor may be configured to perform any or all of the techniques described in this disclosure.

[00164] In the example of FIG. 17, video decoder 300 includes an entropy decoding unit 301, a motion compensation unit 302, an intra prediction unit 303, an inverse quantization unit 304, an inverse transformation unit 305, and a reconstruction unit 306 and a buffer 307. Video decoder 300 may, in some examples, perform a decoding pass generally reciprocal to the encoding pass described with respect to video encoder 200 (FIG. 16).

[00165] Entropy decoding unit 301 may retrieve an encoded bitstream. The encoded bitstream

may include entropy coded video data (e.g., encoded blocks of video data). Entropy decoding unit 301 may decode the entropy coded video data, and from the entropy decoded video data, motion compensation unit 302 may determine motion information including motion vectors, motion vector precision, reference picture list indexes, and other motion information. Motion compensation unit 302 may, for example, determine such information by performing the AMVP and merge mode.

[00166] Motion compensation unit 302 may produce motion compensated blocks, possibly performing interpolation based on interpolation filters. Identifiers for interpolation filters to be used with sub-pixel precision may be included in the syntax elements.

[00167] Motion compensation unit 302 may use interpolation filters as used by video encoder 20 during encoding of the video block to calculate interpolated values for sub-integer pixels of a reference block. Motion compensation unit 302 may determine the interpolation filters used by video encoder 200 according to received syntax information and use the interpolation filters to produce predictive blocks.

[00168] Motion compensation unit 302 may use some of the syntax information to determine sizes of blocks used to encode frame(s) and/or slice(s) of the encoded video sequence, partition information that describes how each macroblock of a picture of the encoded video sequence is partitioned, modes indicating how each partition is encoded, one or more reference frames (and reference frame lists) for each inter-encoded block, and other information to decode the encoded video sequence.

[00169] Intra prediction unit 303 may use intra prediction modes for example received in the bitstream to form a prediction block from spatially adjacent blocks. Inverse quantization unit 303 inverse quantizes, i.e., de-quantizes, the quantized video block coefficients provided in the bitstream and decoded by entropy decoding unit 301. Inverse transform unit 303 applies an inverse transform.

[00170] Reconstruction unit 306 may sum the residual blocks with the corresponding prediction blocks generated by motion compensation unit 202 or intra-prediction unit 303 to form decoded blocks. If desired, a deblocking filter may also be applied to filter the decoded blocks in order to remove blockiness artifacts. The decoded video blocks are then stored in buffer 307, which provides reference blocks for subsequent motion compensation/intra prediction and also

produces decoded video for presentation on a display device.

[00171] In some embodiments, in the ALWIP mode or MIP mode, a prediction block for the current video block is determined by a row and column wise averaging, followed by a matrix multiplication, followed by an interpolation to determine the prediction block.

[00172] FIG. 18 is a flowchart for an example method 1800 of video processing. The method 1800 includes determining 1802, based on an applicability rule, that a differential coding mode is applicable to a conversion between a chroma video block of a video and a bitstream representation of the chroma video block. Operation 1804 includes performing, based on the determining, the conversion between the chroma video block and the bitstream representation of the chroma video block using the differential coding mode, where, in the differential coding mode, the chroma video block is represented in the bitstream representation using a difference between a quantized residual of an intra prediction of the chroma video block and a prediction of the quantized residual, where the intra prediction is performed in a first direction and the prediction of the quantized residual is performed in a second direction, and where, in the differential coding mode, the difference between the quantized residual and the prediction of the quantized residual is represented using a differential pulse coding modulation (DPCM) representation.

[00173] In some embodiments for method 1800, the first direction or the second direction is determined from a vertical direction and a horizontal direction. In some embodiments for method 1800, the intra prediction is performed in the first direction that is same as that used for the intra prediction performed for a luma video block corresponding to the chroma video block. In some embodiments for method 1800, an allowed intra prediction mode for a luma video block is same as that used for performing the intra prediction on the chroma video block corresponding to the luma video block. In some embodiments for method 1800, the second direction allowed for the quantized residual of the chroma video block is same as that used for another quantized residual of a luma video block corresponding to the chroma video block. In some embodiments for method 1800, the second direction used for the prediction of the quantized residual is based on the second direction in which additional quantized residuals are predicted for a luma video block corresponding to the chroma video block. In some embodiments for method 1800, the luma video block is collocated with the chroma video block. In some embodiments for method

1800, the luma video block includes a first sample that is collocated with a second sample located at an upper-left corner of the chroma video block. In some embodiments for method 1800, the luma video block includes a first sample that is collocated with a second sample located at a center of the chroma video block.

[00174] In some embodiments for method 1800, the applicability rule specifies that the differential coding mode is not applied to the chroma video block when the chroma video block is coded with a cross-component linear model (CCLM) coding mode. In some embodiments for method 1800, the applicability rule specifies that the differential coding mode is applied to the chroma video block when the chroma video block is coded with a cross-component linear model (CCLM) coding mode. In some embodiments for method 1800, the applicability rule specifies that the differential coding mode is not applied to the chroma video block when the chroma video block is coded with a joint chroma residual coding technique. In some embodiments for method 1800, the chroma video block is associated with a blue color component or a red color component.

[00175] FIG. 19 is a flowchart for an example method 1900 of video processing. The method 1900 includes performing 1902 a conversion between a current video block of a video and a bitstream representation of the current video block using a differential coding mode and using an intra prediction mode based on a coexistence rule, where, in the intra prediction mode, a quantized residual of an intra prediction of the current video block is based on predictions of samples for the current video block, where, in the differential coding mode, the current video block is represented in the bitstream representing using a difference between the quantized residual and the prediction of the quantized residual, and where the difference between the quantized residual and the prediction of the quantized residual is represented using a differential pulse coding modulation (DPCM) representation.

[00176] In some embodiments for method 1900, the intra prediction mode is a matrix based intra prediction mode (MIP) in which the predictions of the samples for the current video block are determined by performing, on previously coded samples of the video, a boundary downsampling operation, followed by a matrix vector multiplication operation, and selectively followed by an upsampling operation. In some embodiments for method 1900, the coexistence rule restricts the MIP mode to a partial of a plurality of allowed modes for the MIP mode. In

some embodiments for method 1900, the partial of the plurality of allowed modes include a horizontal intra mode and/or a vertical intra mode. In some embodiments for method 1900, the partial of the plurality of allowed modes only includes a horizontal intra mode and/or a vertical intra mode. In some embodiments for method 1900, the coexistence rule indicates that the MIP mode supports all allowed modes for the MIP mode. In some embodiments for method 1900, in the intra prediction mode, the predictions of the samples for the current video block are determined by performing predictions along a non-horizontal direction or a non-vertical direction. In some embodiments for method 1900, the intra prediction mode is a planar or a DC prediction mode. In some embodiments for method 1900, the intra prediction mode is a vertical prediction mode or a horizontal prediction mode. In some embodiments for method 1900, the intra prediction mode is selected from a list of most probable modes (MPM). In some embodiments for method 1900, the intra prediction mode is identified by a field in the bitstream representation. In some embodiments for method 1900, the intra prediction mode is selected from a plurality of allowed intra prediction modes.

[00177] In some embodiments for method 1900, the plurality of allowed intra prediction modes are indicated in a message signaled in: a sequence parameter set (SPS), a video parameter set (VPS), a picture parameter set (PPS), a picture header, a slice header, a tile group header, a largest coding unit (LCU) row, or a group of LCUs. In some embodiments for method 1900, the plurality of allowed intra prediction modes are based on a block dimension of either the current video block or a neighboring video block of the current video block. In some embodiments for method 1900, the plurality of allowed intra prediction modes are based on a shape of either the current video block or a neighboring video block of the current video block. In some embodiments for method 1900, the plurality of allowed intra prediction modes are based on an intra prediction mode of a neighboring video block of the current video block. In some embodiments for method 1900, the plurality of allowed intra prediction modes are based on an inter prediction mode of a neighboring video block of the current video block. In some embodiments for method 1900, the plurality of allowed intra prediction modes are based on an indication of whether a neighboring video block of the current video block is coded using the differential coding mode. In some embodiments for method 1900, the plurality of allowed intra prediction modes are based on a value of a quantization parameter used for the current video

block or for a neighboring video block of the current video block. In some embodiments for method 1900, the plurality of allowed intra prediction modes are based on a color format used for coding the current video block. In some embodiments for method 1900, the plurality of allowed intra prediction mode are based on whether a separate or a dual coding tree structure is used for coding the current video block. In some embodiments for method 1900, the plurality of allowed intra prediction mode are based on a slice or a tile group type or a picture type associated with the current video block.

[00178] In some embodiments for method 1900, in the intra prediction mode, the predictions of the samples for the current video block are determined from non-adjacent samples in a neighboring video block of the current video block. In some embodiments for method 1900, the intra prediction mode comprises an intra block copy merge mode, and wherein the coexistence rule indicates that the differential coding mode is enabled for the IBC merge mode. In some embodiments for method 1900, the intra prediction mode comprises an intra block copy advanced motion vector prediction mode, and wherein the coexistence rule indicates that the differential coding mode is enabled for the intra block copy advanced motion vector prediction mode. In some embodiments for method 1900, the intra prediction mode comprises an intra block copy mode, and a motion vector used in the intra block copy mode and the differential coding mode either is indicated by a field in the bitstream representation, or is derived, or is pre-defined.

[00179] In some embodiments for method 1900, the intra block copy mode is indicated by the motion vector and/or a merge index. In some embodiments for method 1900, the intra block copy mode is indicated by a default motion vector. In some embodiments for method 1900, the default motion vector is $(-w, 0)$, wherein w is a positive integer number. In some embodiments for method 1900, the default motion vector is $(0, -h)$, wherein h is a positive integer number. In some embodiments for method 1900, the default motion vector is $(-w, -h)$, wherein w and h are two positive integer numbers. In some embodiments for method 1900, the field that indicates the motion vector is included in a message signaled in: a sequence parameter set (SPS), a video parameter set (VPS), a picture parameter set (PPS), a picture header, a slice header, a tile group header, a largest coding unit (LCU) row, or a group of LCUs. In some embodiments for method 1900, the motion vector indicated in the field is based on a block dimension of either the current

video block or a neighboring video block of the current video block. In some embodiments for method 1900, the motion vector indicated in the field is based on a shape of either the current video block or a neighboring video block of the current video block. In some embodiments for method 1900, the motion vector indicated in the field is based on an inter prediction mode or an intra prediction mode of a neighboring video block of the current video block. In some embodiments for method 1900, the motion vector indicated in the field is based on motion vectors of a neighboring video block of the current video block. In some embodiments for method 1900, the motion vector indicated in the field is based on an indication of whether a neighboring video block of the current video block is coded using the differential coding mode.

[00180] In some embodiments for method 1900, the motion vector indicated in the field is based on a value of a quantization parameter of the current video block or of a neighboring video block of the current video block. In some embodiments for method 1900, the motion vector indicated in the field is based on a color format used for coding the current video block. In some embodiments for method 1900, the motion vector indicated in the field is based on whether a separate or a dual coding tree structure is used for coding the current video block. In some embodiments for method 1900, the motion vector indicated in the field is based on a slice or a tile group type or a picture type associated with the current video block. In some embodiments for method 1900, in the intra prediction mode, the predictions of the samples for the current video block are determined from non-adjacent samples in a neighboring video block of the current video block using an inter prediction tool.

[00181] In some embodiments for method 1900, the coexistence rule specifies a mapping from a signaled index associated with the differential coding mode to the intra prediction mode. In some embodiments for method 1900, the mapping is indicated in a message signaled in: a sequence parameter set (SPS), a video parameter set (VPS), a picture parameter set (PPS), a picture header, a slice header, a tile group header, a largest coding unit (LCU) row, or a group of LCUs. In some embodiments for method 1900, the mapping is based on a block dimension of either the current video block or a neighboring video block of the current video block. In some embodiments for method 1900, the mapping is based on a shape of either the current video block or a neighboring video block of the current video block. In some embodiments for method 1900, the mapping is based on most probable modes (MPM) of the current video block or of a

neighboring video block of the current video block. In some embodiments for method 1900, the mapping is based on an inter prediction mode or an intra prediction mode of a neighboring video block of the current video block. In some embodiments for method 1900, the mapping is based on motion vectors of a neighboring video block of the current video block. In some embodiments for method 1900, the mapping is based on an indication of whether a neighboring video block of the current video block is coded using the differential coding mode. In some embodiments for method 1900, the mapping is based on a value of a quantization parameter of the current video block or of a neighboring video block of the current video block. In some embodiments for method 1900, the mapping is based on a color format used for coding the current video block. In some embodiments for method 1900, the mapping is based on whether a separate or a dual coding tree structure is used for coding the current video block. In some embodiments for method 1900, the mapping is based on a transform type applied to the current video block. In some embodiments for method 1900, the mapping is based on a slice or a tile group type or a picture type associated with the current video block.

[00182] In some embodiments for method 1900, the prediction of the quantized residual is performed in a residual prediction direction that is different from a horizontal direction or a vertical direction. In some embodiments for method 1900, the residual prediction direction is a 45-degree direction. In some embodiments for method 1900, where $Q(r_{i,j})$ indicates the quantized residual, where $\tilde{r}_{i,j}$ indicates residual quantized samples, and where $\tilde{r}_{i,j}$ may be derived by $Q(r_{i,j}) - Q(r_{(i-1),(j-1)})$ in response to the $Q(r_{(i-1),(j-1)})$ being available. In some embodiments for method 1900, the residual prediction direction is a 135-degree direction. In some embodiments for method 1900, where $Q(r_{i,j})$ indicates the quantized residual, where $\tilde{r}_{i,j}$ indicates residual quantized samples, and where $\tilde{r}_{i,j}$ may be derived by $Q(r_{i,j}) - Q(r_{(i-1),(j+1)})$ in response to the $Q(r_{(i-1),(j+1)})$ being available.

[00183] In some embodiments for method 1900, the prediction of the quantized residual is performed in a residual prediction direction that is based on a first value and a second value. In some embodiments for method 1900, where $Q(r_{i,j})$ indicates the quantized residual, where $\tilde{r}_{i,j}$ indicates residual quantized samples, where $\tilde{r}_{i,j}$ may be derived by $Q(r_{i,j}) - Q(r_{(i-m),(j-n)})$ in response to the $Q(r_{(i-m),(j-n)})$ being available, where m is the first value and n is the second value, where i is a third value indicative of a first position along a plurality of rows of the current

video block, and where j is a fourth value indicative of a second position along a plurality of columns of the current video block. In some embodiments for method 1900, m and n are signaled in the bitstream representation. In some embodiments for method 1900, m and n are integers.

[00184] In some embodiments for method 1900, m and n are indicated in a message included in: a sequence parameter set (SPS), a video parameter set (VPS), a picture parameter set (PPS), a picture header, a slice header, a tile group header, a largest coding unit (LCU) row, or a group of LCUs. In some embodiments for method 1900, m is based on the third value for i and n is based on the fourth value for j . In some embodiments for method 1900, the first value for m and the second value for n are based on a block dimension of either the current video block or a neighboring video block of the current video block. In some embodiments for method 1900, the first value for m and the second value for n are based on a shape of either the current video block or a neighboring video block of the current video block. In some embodiments for method 1900, the first value for m and the second value for n are based on most probable modes (MPM) of the current video block or of a neighboring video block of the current video block. In some embodiments for method 1900, the first value for m and the second value for n are based on an inter prediction mode or an intra prediction mode of a neighboring video block of the current video block. In some embodiments for method 1900, the first value for m and the second value for n are based on motion vectors of a neighboring video block of the current video block. In some embodiments for method 1900, the first value for m and the second value for n are based on an indication of whether a neighboring video block of the current video block is coded using the differential coding mode. In some embodiments for method 1900, the first value for m and the second value for n are based on a value of a quantization parameter of the current video block or of a neighboring video block of the current video block. In some embodiments for method 1900, the first value for m and the second value for n are based on a color format used for coding the current video block. In some embodiments for method 1900, the first value for m and the second value for n are based on whether a separate or a dual coding tree structure is used for coding the current video block. In some embodiments for method 1900, the first value for m and the second value for n are based on a slice or a tile group type or a picture type associated with the current video block.

[00185] FIG. 20 is a flowchart for an example method 2000 of video processing. The method 2000 includes performing 2002 a conversion between a current video block of a video and a bitstream representation of the current video block using a differential coding mode in which the current video block is represented in the bitstream representation using a difference between a quantized residual of an intra prediction of the current video block and a prediction of the quantized residual, where a direction of the prediction of the quantized residual is inferable from the bitstream representation.

[00186] In some embodiments for method 2000, the direction of the prediction of the quantized residual is implicitly inferable from an intra prediction mode used for the intra prediction. In some embodiments for method 2000, the direction of the prediction of the quantized residual is inferable to be a same direction as that used for the intra prediction mode. In some embodiments for method 2000, the direction is vertical or horizontal. In some embodiments for method 2000, the direction of the prediction of the quantized residual is inferable to be a first direction that is different from a second direction used for the intra prediction mode. In some embodiments for method 2000, the first direction is vertical and the second direction is horizontal. In some embodiments for method 2000, the first direction is horizontal and the second direction is vertical.

[00187] In some embodiments for method 2000, the direction of the prediction of the quantized residual is inferable based on a message signaled in: a sequence parameter set (SPS), a video parameter set (VPS), a picture parameter set (PPS), a picture header, a slice header, a tile group header, a largest coding unit (LCU) row, or a group of LCUs. In some embodiments for method 2000, the direction of the prediction of the quantized residual is inferable based on a block dimension of either the current video block or a neighboring video block of the current video block. In some embodiments for method 2000, the direction of the prediction of the quantized residual is inferable based on a shape of either the current video block or a neighboring video block of the current video block. In some embodiments for method 2000, the direction of the prediction of the quantized residual is inferable based on most probable modes (MPM) of the current video block or of a neighboring video block of the current video block. In some embodiments for method 2000, the direction of the prediction of the quantized residual is inferable based on an inter prediction mode or an intra prediction mode of a neighboring video

block of the current video block. In some embodiments for method 2000, the direction of the prediction of the quantized residual is inferable based on motion vectors of a neighboring video block of the current video block. In some embodiments for method 2000, the direction of the prediction of the quantized residual is inferable based on an indication of whether a neighboring video block of the current video block is coded using the differential coding mode. In some embodiments for method 2000, the direction of the prediction of the quantized residual is inferable based on a value of a quantization parameter of the current video block or of a neighboring video block of the current video block. In some embodiments for method 2000, the direction of the prediction of the quantized residual is inferable based on a color format used for coding the current video block. In some embodiments for method 2000, the direction of the prediction of the quantized residual is inferable based on whether a separate or a dual coding tree structure is used for coding the current video block. In some embodiments for method 2000, the direction of the prediction of the quantized residual is inferable based on a transform type applied to the current video block. In some embodiments for method 2000, the direction of the prediction of the quantized residual is inferable based on a slice or a tile group type or a picture type associated with the current video block.

[00188] FIG. 21 is a flowchart for an example method 2100 of video processing. The method 2100 includes determining 2102 that a differential coding mode is applicable to a conversion between a current video block of a video and a bitstream representation of the current video block, and performing 2104, based on the determining, the conversion between the current video block and the bitstream representation of the current video block using the differential coding mode according to an implementation rule, where, in the differential coding mode, the current video block is represented in the bitstream representation using a difference between a quantized residual of an intra prediction of the current video block and a prediction of the quantized residual, and where, in the differential coding mode, the difference between the quantized residual and the prediction of the quantized residual is represented using a differential pulse coding modulation (DPCM) representation.

[00189] In some embodiments for method 2100, the implementation rule specifies that values of the quantized residual are restricted within a range of values. In some embodiments for method 2100, the implementation rule specifies that values of the difference between the quantized

residual and the prediction of the quantized residual are restricted within a range of values. In some embodiments for method 2100, the implementation rule specifies that a clipping operation is applied to the difference between the quantized residual and the prediction of the quantized residual, and the clipping operation restricts values of the difference between the quantized residual and the prediction of the quantized residual within a range of values. In some embodiments for method 2100, the implementation rule specifies that a clipping operation is applied to the quantized residual, and the clipping operation restricts values of the quantized residual within a range of values.

[00190] In some embodiments for method 2100, the difference between the quantized residual and the prediction of the quantized residual is defined according to any one of the following

$$\text{equations: } \tilde{r}_{i,j} = \begin{cases} Q(r_{i,j}), & i = 0, \quad 0 \leq j \leq (N-1) \\ Q(r_{i,j}) - Q(r_{(i-1),j}), & 1 \leq i \leq (M-1), \quad 0 \leq j \leq (N-1) \end{cases}, \text{ and } \tilde{r}_{i,j} = \begin{cases} Q(r_{i,j}), & 0 \leq i \leq (M-1), \quad j = 0 \\ Q(r_{i,j}) - Q(r_{i,(j-1)}), & 0 \leq i \leq (M-1), \quad 1 \leq j \leq (N-1) \end{cases}, \text{ where } \tilde{r}_{i,j} \text{ describes the}$$

difference between the quantized residual and the prediction of the quantized residual,

wherein $Q(r_{i,j})$ describes the values of the quantized residual, where M and N respectively describe rows and columns of the current video block, and where i and j respectively describe positions along the rows and columns of the current video block.

[00191] In some embodiments for method 2100, the quantized residual is defined according to any one of the following equations: $Q(r_{i,j}) = \sum_{k=0}^i \tilde{r}_{k,j}$, $0 \leq i \leq (M-1)$, $0 \leq j \leq (N-1)$, and $Q(r_{i,j}) = \sum_{k=0}^j \tilde{r}_{i,k}$, $0 \leq i \leq (M-1)$, $0 \leq j \leq (N-1)$, where $\tilde{r}_{k,j}$ and $\tilde{r}_{i,k}$ describe the difference between the quantized residual and the prediction of the quantized residual,

wherein $Q(r_{i,j})$ describes the quantized residual, where M and N respectively describe rows and columns of the current video block, and where i and j respectively describe positions along the rows and columns of the current video block.

[00192] In some embodiments for method 2100, each value from the values is restricted by the clipping operation according to the following clipping rule: a value is a minimum value in response to the value being less than the minimum value, the value is a maximum value in response to the value being greater than the maximum value. In some embodiments for method 2100, each value from the values is restricted by the clipping operation according to the

following clipping rule: a value is a minimum value in response to the value being less than or equal to the minimum value, and the value is a maximum value in response to the value being greater than or equal to the maximum value. In some embodiments for method 2100, each value from the values is restricted by the clipping operation according to the following clipping rule: a value is a minimum value in response to the value being less than the minimum value, and the value is a maximum value in response to the value being greater than or equal to the maximum value. In some embodiments for method 2100, each value from the values is restricted by the clipping operation according to the following clipping rule: a value is a minimum value in response to the value being less than or equal to the minimum value, and the value is a maximum value in response to the value being greater than the maximum value.

[00193] In some embodiments for method 2100, the minimum value and/or the maximum value is or are a positive value or a negative value. In some embodiments for method 2100, the minimum value is -32768 and the maximum value is 32767. In some embodiments for method 2100, the minimum value and/or the maximum value are based on another range of values related to inverse quantization for a video block that is not coded with the differential coding mode. In some embodiments for method 2100, the minimum value and/or the maximum value are based on a bit depth of an input sample of the quantized residual block or a reconstructed sample of the reconstructed quantized residual block. In some embodiments for method 2100, the minimum value and/or the maximum value are based on whether the current video block is encoded or decoded with a lossless coding technique. In some embodiments for method 2100, the minimum value and/or the maximum value are based on an indication of whether a transquant bypass flag is enabled, wherein whether the lossless coding technique is used is indicated by the transquant bypass flag. In some embodiments for method 2100, the minimum value and/or the maximum value are based on an indication of whether a coding unit (CU) transquant bypass flag is enabled, wherein whether the lossless coding technique is used at a CU level is indicated by the CU transquant bypass flag. In some embodiments for method 2100, the minimum value and/or the maximum value are based on a message signaled in: a sequence parameter set (SPS), a video parameter set (VPS), a picture parameter set (PPS), a picture header, a slice header, a tile group header, a largest coding unit (LCU) row, or a group of LCUs.

[00194] In some embodiments for method 2100, the minimum value and/or the maximum value

are based on a block dimension of either the current video block or a neighboring video block of the current video block. In some embodiments for method 2100, the minimum value and/or the maximum value are based on a shape of either the current video block or a neighboring video block of the current video block. In some embodiments for method 2100, the minimum value and/or the maximum value are based on most probable modes (MPM) of the current video block or of a neighboring video block of the current video block. In some embodiments for method 2100, the minimum value and/or the maximum value are based on an inter prediction mode or an intra prediction mode of a neighboring video block of the current video block. In some embodiments for method 2100, the minimum value and/or the maximum value are based on motion vectors of a neighboring video block of the current video block. In some embodiments for method 2100, the minimum value and/or the maximum value are based on an indication of whether a neighboring video block of the current video block is coded using the differential coding mode. In some embodiments for method 2100, the minimum value and/or the maximum value are based on a value of a quantization parameter of the current video block or of a neighboring video block of the current video block. In some embodiments for method 2100, the minimum value and/or the maximum value are based on a color format used for coding the current video block. In some embodiments for method 2100, the minimum value and/or the maximum value are based on whether a separate or a dual coding tree structure is used for coding the current video block.

[00195] In some embodiments for method 2100, the minimum value and/or the maximum value are based on a transform type applied to the current video block. In some embodiments for method 2100, the minimum value and/or the maximum value are based on a slice or a tile group type or a picture type associated with the current video block. In some embodiments for method 2100, the implementation rule specifies that prediction of the quantized residual is performed in a direction from a last row or last column of the current video block to a first row or a first column of the current video block. In some embodiments for method 2100, in response to the direction being horizontal, the prediction of a value of the quantized residual at an i -th column of the current video block is the value of the quantized residual at the $(i+1)$ -th column of the current video block. In some embodiments for method 2100, in response to the direction being vertical, the prediction of a value of the quantized residual at an i -th row of the current video block is the

value of the quantized residual at the (i+1)-th row of the current video block.

[00196] In some embodiments for method 2100, the implementation rule specifies that the differential coding mode is applied to a subset or a portion of the current video block. In some embodiments for method 2100, the implementation rule specifies that the differential coding mode is not applied to left-most k columns of the current video block when the prediction of the quantized residual is performed in a horizontal direction. In some embodiments for method 2100, the implementation rule specifies that the differential coding mode is not applied to upper-most k rows of the current video block when the prediction of the quantized residual is performed in a vertical direction. In some embodiments for method 2100, the implementation rule specifies that the differential coding mode is not applied to right-most k columns of the current video block when the prediction of the quantized residual is performed in a horizontal direction. In some embodiments for method 2100, the implementation rule specifies that the differential coding mode is not applied to bottom-most k rows of the current video block when the prediction of the quantized residual is performed in a vertical direction. In some embodiments for method 2100, a value for k is predefined.

[00197] In some embodiments for method 2100, a value for k is indicated in a message signaled in: a sequence parameter set (SPS), a video parameter set (VPS), a picture parameter set (PPS), a picture header, a slice header, a tile group header, a largest coding unit (LCU) row, or a group of LCUs. In some embodiments for method 2100, a value for k is based on a block dimension of either the current video block or a neighboring video block of the current video block. In some embodiments for method 2100, a value for k is based on a shape of either the current video block or a neighboring video block of the current video block. In some embodiments for method 2100, a value for k is based on most probable modes (MPM) of the current video block or of a neighboring video block of the current video block. In some embodiments for method 2100, a value for k is based on an inter prediction mode or an intra prediction mode of a neighboring video block of the current video block. In some embodiments for method 2100, a value for k is based on an intra prediction mode of the current video block. In some embodiments for method 2100, a value for k is based on motion vectors of a neighboring video block of the current video block. In some embodiments for method 2100, a value for k is based on an indication of whether a neighboring video block of the current video block is coded using the differential coding mode.

[00198] In some embodiments for method 2100, a value for k is based on a value of a quantization parameter of the current video block or of a neighboring video block of the current video block. In some embodiments for method 2100, a value for k is based on a color format used for coding the current video block. In some embodiments for method 2100, a value for k is based on whether a separate or a dual coding tree structure is used for coding the current video block. In some embodiments for method 2100, a value for k is based on a transform type applied to the current video block. In some embodiments for method 2100, a value for k is based on a slice or a tile group type or a picture type associated with the current video block.

[00199] In some embodiments for method 2100, the implementation rule specifies that the differential coding mode is applied on a segment-by-segment basis on the current video block. In some embodiments for method 2100, the implementation rule specifies that residual quantized samples are determined according to the following formula when the prediction of the quantized residual is performed in a vertical direction and when $N=nK$:

$$\tilde{r}_{i,j} = \begin{cases} Q(r_{i,j}), & i \% K = 0, \quad 0 \leq j \leq (N - 1) \\ Q(r_{i,j}) - Q(r_{(i-1),j}), & i \% K > 0, \quad i \leq (M - 1), \quad 0 \leq j \leq (N - 1) \end{cases},$$

where $\tilde{r}_{i,j}$ describes the difference between the quantized residual and the prediction of the quantized residual, where $Q(r_{i,j})$ describes the quantized residual, where M and N respectively describe rows and columns of the current video block, where i and j respectively describe positions along the rows and columns of the current video block, and where n and K are integers.

[00200] In some embodiments for method 2100, the implementation rule specifies that residual quantized samples are determined according to the following formula when the prediction of the quantized residual is performed in a horizontal direction and when $M=mK$:

$$\tilde{r}_{i,j} = \begin{cases} Q(r_{i,j}), & 0 \leq i \leq (M - 1), \quad j \% K = 0 \\ Q(r_{i,j}) - Q(r_{i,(j-1)}), & 0 \leq i \leq (M - 1), \quad j \% K > 0, \quad j \leq (N - 1) \end{cases},$$

where $\tilde{r}_{i,j}$ describes residual quantized samples, where $Q(r_{i,j})$ describes the quantized residual, where M and N respectively describe rows and columns of the current video block, where i and j respectively describe positions along the rows and columns of the current video block, and where m and K are integers.

[00201] FIG. 22 is a flowchart for an example method 2200 of video processing. The method 2200 includes determining 2202, based on an applicability rule, that a differential coding mode is

applicable to a conversion between a current video block of a video and a bitstream representation of the current video block, where the current video block is associated with a color component, and where the applicability rule specifies that whether the differential coding mode is enabled or disabled for the current video block based on whether the differential coding mode is applied to another video block of the video. The method 2200 also includes performing 2204, based on the determining, the conversion between the current video block and the bitstream representation of the current video block using the differential coding mode, where, in the differential coding mode, the current video block is represented in the bitstream representation using a difference between a quantized residual of an intra prediction of the current video block and a prediction of the quantized residual, and where, in the differential coding mode, the difference between the quantized residual and the prediction of the quantized residual is represented using a differential pulse coding modulation (DPCM) representation.

[00202] In some embodiments for method 2200, the current video block is a chroma video block, the applicability rule specifies that whether the differential coding mode is enabled for the chroma video block is based on whether the differential coding mode is applied to one or more representative luma video blocks, and the one or more representative luma video blocks are collocated with the chroma video block. In some embodiments for method 2200, the one or more representative luma video blocks are defined in a same manner as that used for deriving a derived mode for the intra prediction for the chroma video block. In some embodiments for method 2200, the current video block is a chroma video block, the applicability rule specifies that the differential coding mode is enabled for the chroma video block when a derived mode used for the intra prediction of the chroma video block and the differential coding mode is enabled for one or more representative luma blocks, and the one or more representative luma video blocks are collocated with the chroma video block. In some embodiments for method 2200, the applicability rule specifies that whether the differential coding mode is enabled or disabled for the current video block is indicated in one or more syntax elements included in the bitstream representation.

[00203] In some embodiments for method 2200, a single syntax element indicates whether the differential coding mode is enabled or disabled for two color components, and the two color components include the color component associated with the current video block. In some

embodiments for method 2200, two syntax elements indicate whether the differential coding mode is enabled or disabled for two color components, respectively, and the two color components include the color component associated with the current video block. In some embodiments for method 2200, the applicability rule specifies that the bitstream representation excludes a syntax element that indicates whether the differential coding mode is enabled or disabled for the current video block when a specific mode used for the intra prediction of the current video block. In some embodiments for method 2200, the specific mode includes a cross-component linear model (CCLM).

[00204] The following listing of examples is a description of additional embodiments.

[00205] 1. A method of video processing, comprising: performing a conversion between a current video block and a bitstream representation of the current video block using a differential coding mode and selectively using an intra prediction mode based on a coexistence rule; wherein the intra prediction mode is used for generating predictions for samples of the current video block; and wherein, the differential coding mode is used to represent a quantized residual block from the predictions of the pixels, using a differential pulse coding modulation representation.

[00206] 2. The method of example 1, wherein the intra prediction mode is a matrix based intra prediction mode (MIP), and wherein the coexistence rule restricts the MIP to a partial of allowed modes of the MIP.

[00207] 3. The method of example 2, wherein the partial of allowed modes include horizontal or vertical normal intra modes.

[00208] Further embodiments of examples 1-3 are described in item 1, section 4. For example, the differential coding mode may represent the current version of the QR-BDPCM coding mode.

[00209] 4. The method of example 1, wherein the intra prediction mode includes a prediction along a non-horizontal or a non-vertical direction.

[00210] 5. The method of example 1 or 4, wherein the intra prediction mode is a planar or a DC prediction mode.

[00211] 6. The method of example 1 or 4, wherein the intra prediction modes is a vertical or a horizontal prediction mode.

[00212] 7. The method of example 1 or 4, wherein the intra prediction mode is identified by a field in the bitstream representation.

- [00213] 8. The method of example 1 or 4, wherein the intra prediction mode is dependent on a block dimension of the current video block or a neighboring block.
- [00214] 9. The method of example 1 or 4, wherein the intra prediction mode is dependent on a shape of the current block or a neighboring block.
- [00215] 10. The method of example 1 or 4, wherein the intra prediction mode is dependent on whether the current video block or the neighboring video block is coded using inter prediction or intra prediction.
- [00216] 11. The method of example 1 or 4, wherein the intra prediction mode is dependent on whether the neighboring video block is coded using the differential coding mode.
- [00217] 12. The method of example 1 or 4, wherein the intra prediction mode is dependent on value of a quantization parameter used for the current video block or the neighboring video block.
- [00218] 13. The method of example 1 or 4, wherein the intra prediction mode is dependent on a color format used for coding the current video block.
- [00219] 14. The method of example 1 or 4, wherein the intra prediction mode is dependent on whether a separate or a dual coding tree structure is used for coding the current video block.
- [00220] Further embodiments of examples 4 to 14 are provided in item 2 of section 4.
- [00221] 15. The method of example 1, wherein the generating predictions for samples of the current video block is performed from non-adjacent samples in a neighboring video region.
- [00222] 16. The method of example 1, wherein the intra prediction mode comprises an intra block copy merge mode.
- [00223] 17. The method of example 1, wherein the intra prediction mode comprises an intra block copy advanced motion vector prediction mode.
- [00224] 18. The method of any of examples 15 to 17, wherein the intra prediction mode is indicated by a block vector or a merge index.
- [00225] Further embodiments of examples 15 to 18 are provided in item 3 of section 4.
- [00226] 19. The method of example 1, wherein the coexistence rule specifies a mapping from a signaled index in the differential coding mode to the intra prediction mode based on a field in the bitstream representation.
- [00227] 20. The method of example 1, wherein the coexistence rule specifies a mapping from

a signaled index in the differential coding mode to the intra prediction mode based on a dimension of the current video block or a neighboring block.

[00228] 21. The method of example 1, wherein the coexistence rule specifies a mapping from a signaled index in the differential coding mode to the intra prediction mode based on a shape of the current video block or a neighboring block.

[00229] 22. The method of example 1, wherein the coexistence rule specifies a mapping from a signaled index in the differential coding mode to the intra prediction mode based on a prediction mode of the current video block or a neighboring block.

[00230] 23. The method of example 1, wherein the coexistence rule specifies a mapping from a signaled index in the differential coding mode to the intra prediction mode based on most probable modes of the current video block or a neighboring block.

[00231] 24. The method of example 1, wherein the coexistence rule specifies a mapping from a signaled index in the differential coding mode to the intra prediction mode based on motion vectors of the current video block or a neighboring block.

[00232] 25. The method of example 1, wherein the coexistence rule specifies a mapping from a signaled index in the differential coding mode to the intra prediction mode based on whether or not a neighboring block is coded using the differential coding mode.

[00233] 26. The method of example 1, wherein the coexistence rule specifies a mapping from a signaled index in the differential coding mode to the intra prediction mode based on a quantization parameter used by the current video block or a neighboring block.

[00234] 27. The method of example 1, wherein the coexistence rule specifies a mapping from a signaled index in the differential coding mode to the intra prediction mode based on a color format of the current video block.

[00235] 28. The method of example 1, wherein the coexistence rule specifies a mapping from a signaled index in the differential coding mode to the intra prediction mode based on whether a separate or a dual coding tree is used by the current video block.

[00236] 29. The method of example 1, wherein the coexistence rule specifies a mapping from a signaled index in the differential coding mode to the intra prediction mode based on a transform applied to the current video block.

[00237] 30. The method of example 1, wherein the coexistence rule specifies a mapping from a

signaled index in the differential coding mode to the intra prediction mode based on a slice type or a tile group type or a picture type of the current video block.

[00238] Further embodiments of examples 19 to 30 are provided in item 2 of section 4.

[00239] 31. A method of video processing, comprising: performing a conversion between a current video block and a bitstream representation of the current video block using a differential coding mode in which a quantized residual block from a predictions of pixels of the current video block is represented using a differential pulse coding modulation representation; wherein a first direction of the prediction or a second direction of the differential coding mode is inferable from the bitstream representation.

[00240] 32. The method of example 31, wherein the first direction of the prediction of pixels is implicitly inferable from an intra prediction mode used for the predicting.

[00241] 33. The method of example 32, wherein the second direction of the differential coding mode is inferable to be a same direction as the first direction of the prediction.

[00242] 34. The method of example 31, wherein the second direction is inferable from an intra prediction mode used for the predicting.

[00243] 35. The method of example 31, wherein the second direction is inferable from a dimension of the current video block or a neighboring block or a shape of the current video block or a neighboring block.

[00244] 36. The method of example 31, wherein the second direction is inferable from motion vectors of a neighboring block.

[00245] 37. The method of example 31, wherein the second direction is inferable from most probable modes of the current video block or a neighboring block.

[00246] 38. The method of example 31, wherein the second direction is inferable from a prediction mode of a neighboring block.

[00247] 39. The method of example 31, wherein the second direction is inferable from an intra prediction mode of a neighboring block.

[00248] 40. The method of example 31, wherein the second direction is inferable from whether or not a neighboring block uses the differential coding mode.

[00249] Further embodiments of examples 31-40 are provided in item 4 of section 4.

[00250] 41. A method of video processing, comprising: determining, based on an applicability

rule, that a differential coding mode is applicable to a conversion between a current video block and a bitstream representation of the current video block; and performing the conversion between a current video block and a bitstream representation using the differential coding mode; wherein, in the differential coding mode, a quantized residual block from intra prediction of pixels of the current video block is represented using a differential pulse coding modulation representation performed in a residual prediction direction that is different from a horizontal or a vertical direction.

[00251] 42. The method of example 41, wherein the residual prediction direction is a 45-degree direction.

[00252] 43. The method of example 41, wherein the residual prediction direction is a 135-degree direction.

[00253] 44. The method of example 41, wherein the residual prediction direction is related to a field in the bitstream representation or a dimension of the current video block or a neighboring block or a shape of the current video block or the neighboring block.

[00254] Further embodiments of examples 41 to 44 are provided in item 6 of section 4.

[00255] 45. The method of example 41, wherein the applicability rule specifies to use the differential coding mode due to the current video block being a chroma block.

[00256] 46. The method of example 45, wherein the applicability rule further specifies to that the residual prediction direction for the current video block is a same direction as that for a luma block corresponding to the current video block.

[00257] 47. The method of example 41, wherein the applicability rule specifies to use the differential coding due to the current video block not using a cross-component linear model (CCLM) coding mode.

[00258] Further embodiments of examples 45 to 47 are provided in item 7 of section 4.

[00259] 48. The method of example 41, wherein the applicability rule specifies to derive applicability of the differential coding mode for one color component from applicability of the differential coding mode for another color component.

[00260] Further embodiments of example 48 are provided in item 12 of section 4.

[00261] 49. A method of video processing, comprising: determining that a differential coding mode is applicable to a conversion between a current video block and a bitstream representation

of the current video block; and performing the conversion between a current video block and a bitstream representation using an implementation rule of the differential coding mode; wherein, in the differential coding mode, a quantized residual block from intra prediction of pixels of the current video block is represented using a differential pulse coding modulation representation performed in a residual prediction direction that is different from a horizontal or a vertical direction.

[00262] 50. The method of example 49, wherein the implementation rule specifies to restrict values of the quantized residual block within a range.

[00263] 51. The method of example 49, wherein the implementation rule specifies using clipping to obtain the quantized residual block.

[00264] Further embodiments of examples 49-51 are provided in item 8 of section 4.

[00265] 52. The method of example 49, wherein the implementation rule specifies to perform prediction from a last row of the current video block to a first row of the current video block.

[00266] 53. The method of example 49, wherein the implementation rule specifies to perform prediction from a last column of the current video block to a first column of the current video block.

[00267] Further embodiments of examples 52 to 53 are provided in item 9 of section 4.

[00268] 54. The method of example 49, wherein the implementation rule specifies to apply the differential coding mode to only a subset of the current video block.

[00269] 55. The method of example 54, wherein the subset excludes k left columns of residues, where k is an integer smaller than a pixel width of the block.

[00270] 56. The method of example 54, wherein the subset excludes k top rows of residues, where k is an integer smaller than a pixel height of the block.

[00271] Further embodiments of examples 54 to 56 are provided in item 10 of section 4.

[00272] 57. The method of example 49, wherein the implementation rule specifies to apply the differential coding mode on a segment by segment basis to the conversion.

[00273] Further embodiments of example 57 are provided in item 11 of section 4.

[00274] 58. A video processing apparatus comprising a processor configured to implement one or more of examples 1 to 57.

[00275] 59. A computer-readable medium having code stored thereon, the code, when executed

by a processor, causing the processor to implement a method recited in any one or more of examples 1 to 57.

[00276] In the listing of examples in this present document, the term conversion may refer to the generation of the bitstream representation for the current video block or generating the current video block from the bitstream representation. The bitstream representation need not represent a contiguous group of bits and may be divided into bits that are included in header fields or in codewords representing coded pixel value information.

[00277] In the examples above, the applicability rule may be pre-defined and known to encoders and decoders.

[00278] It will be appreciated that the disclosed techniques may be embodied in video encoders or decoders to improve compression efficiency using techniques that include the use of various implementation rules of considerations regarding the use of a differential coding mode in intra coding, as described in the present document.

[00279] The disclosed and other solutions, examples, embodiments, modules and the functional operations described in this document can be implemented in digital electronic circuitry, or in computer software, firmware, or hardware, including the structures disclosed in this document and their structural equivalents, or in combinations of one or more of them. The disclosed and other embodiments can be implemented as one or more computer program products, i.e., one or more modules of computer program instructions encoded on a computer readable medium for execution by, or to control the operation of, data processing apparatus. The computer readable medium can be a machine-readable storage device, a machine-readable storage substrate, a memory device, a composition of matter effecting a machine-readable propagated signal, or a combination of one or more them. The term “data processing apparatus” encompasses all apparatus, devices, and machines for processing data, including by way of example a programmable processor, a computer, or multiple processors or computers. The apparatus can include, in addition to hardware, code that creates an execution environment for the computer program in question, e.g., code that constitutes processor firmware, a protocol stack, a database management system, an operating system, or a combination of one or more of them. A propagated signal is an artificially generated signal, e.g., a machine-generated electrical, optical, or electromagnetic signal, that is generated to encode information for

transmission to suitable receiver apparatus.

[00280] A computer program (also known as a program, software, software application, script, or code) can be written in any form of programming language, including compiled or interpreted languages, and it can be deployed in any form, including as a stand-alone program or as a module, component, subroutine, or other unit suitable for use in a computing environment. A computer program does not necessarily correspond to a file in a file system. A program can be stored in a portion of a file that holds other programs or data (e.g., one or more scripts stored in a markup language document), in a single file dedicated to the program in question, or in multiple coordinated files (e.g., files that store one or more modules, sub programs, or portions of code). A computer program can be deployed to be executed on one computer or on multiple computers that are located at one site or distributed across multiple sites and interconnected by a communication network.

[00281] The processes and logic flows described in this document can be performed by one or more programmable processors executing one or more computer programs to perform functions by operating on input data and generating output. The processes and logic flows can also be performed by, and apparatus can also be implemented as, special purpose logic circuitry, e.g., an FPGA (field programmable gate array) or an ASIC (application specific integrated circuit).

[00282] Processors suitable for the execution of a computer program include, by way of example, both general and special purpose microprocessors, and any one or more processors of any kind of digital computer. Generally, a processor will receive instructions and data from a read only memory or a random-access memory or both. The essential elements of a computer are a processor for performing instructions and one or more memory devices for storing instructions and data. Generally, a computer will also include, or be operatively coupled to receive data from or transfer data to, or both, one or more mass storage devices for storing data, e.g., magnetic, magneto optical disks, or optical disks. However, a computer need not have such devices. Computer readable media suitable for storing computer program instructions and data include all forms of non-volatile memory, media and memory devices, including by way of example semiconductor memory devices, e.g., EPROM, EEPROM, and flash memory devices; magnetic disks, e.g., internal hard disks or removable disks; magneto optical disks; and CD

ROM and DVD-ROM disks. The processor and the memory can be supplemented by, or incorporated in, special purpose logic circuitry.

[00283] While this patent document contains many specifics, these should not be construed as limitations on the scope of any subject matter or of what may be claimed, but rather as descriptions of features that may be specific to particular embodiments of particular techniques. Certain features that are described in this patent document in the context of separate embodiments can also be implemented in combination in a single embodiment. Conversely, various features that are described in the context of a single embodiment can also be implemented in multiple embodiments separately or in any suitable subcombination. Moreover, although features may be described above as acting in certain combinations and even initially claimed as such, one or more features from a claimed combination can in some cases be excised from the combination, and the claimed combination may be directed to a subcombination or variation of a subcombination.

[00284] Similarly, while operations are depicted in the drawings in a particular order, this should not be understood as requiring that such operations be performed in the particular order shown or in sequential order, or that all illustrated operations be performed, to achieve desirable results. Moreover, the separation of various system components in the embodiments described in this patent document should not be understood as requiring such separation in all embodiments.

[00285] Only a few implementations and examples are described and other implementations, enhancements and variations can be made based on what is described and illustrated in this patent document.

CLAIMS

1. A method of processing video data, comprising:
 - determining, for a conversion between a current video block of a video and a bitstream of the video, that a differential coding mode used for residual samples is applied to the current video block; and
 - performing, based on the determining, the conversion using the differential coding mode, wherein, in the differential coding mode, differences between quantized residuals derived with an intra prediction and predictors of the quantized residuals are presented in the bitstream, and wherein at least one of reconstructed quantized residuals is derived based on a sum of one difference and another reconstructed quantized residual, wherein values of the reconstructed quantized residuals are restricted within a specific range by applying a clipping operation to the reconstructed quantized residuals, and wherein a single syntax element indicates whether the differential coding mode is applied to two chroma components or not, and the single syntax element is excluded from the bitstream in response to a specific mode being used for the intra prediction of two chroma components.
2. The method of claim 1, wherein each of the values of the reconstructed quantized residuals is restricted by the clipping operation as following:
 - a value is a minimum value in response to the value being less than the minimum value, and
 - the value is a maximum value in response to the value being greater than the maximum value.
3. The method of claim 2, wherein the minimum value is a negative value, and the maximum value is a positive value.
4. The method of claim 3, wherein the minimum value is set to -32768, and the maximum value is set to 32767.

5. The method of claim 1, wherein whether the differential coding mode is applied to the current video block or not is indicated by one or more syntax elements included in the bitstream.
6. The method of claim 1, wherein the specific mode includes a cross-component linear model.
7. The method of claim 1, wherein the predictors of the quantized residuals are generated at sample level.
8. The method of claim 7, wherein the differential coding mode applied along a horizontal or vertical direction.
9. The method of claim 1, wherein the conversion includes encoding the current video block into the bitstream.
10. The method of claim 1, wherein the conversion includes decoding the current video block from the bitstream.
11. An apparatus for processing video data comprising a processor and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor, cause the processor to:
 - determine, for a conversion between a current video block of a video and a bitstream of the video, that a differential coding mode used for residual samples is applied to the current video block; and
 - perform, based on the determining, the conversion using the differential coding mode, wherein, in the differential coding mode, differences between quantized residuals derived with an intra prediction and predictors of the quantized residuals are presented in the bitstream, and wherein at least one of reconstructed quantized residuals is derived based on a sum of one difference and another reconstructed quantized residual,
 - wherein values of the reconstructed quantized residuals are restricted within a specific range by a clipping operation applied to the reconstructed quantized residuals, and

wherein a single syntax element indicates whether the differential coding mode is applied to two chroma components or not, and the single syntax element is excluded from the bitstream in response to a specific mode being used for the intra prediction of two chroma components.

12. The apparatus of claim 11, wherein each of the values of the reconstructed quantized residuals is restricted by the clipping operation as following:

a value is a minimum value in response to the value being less than the minimum value, and the value is a maximum value in response to the value being greater than the maximum value, and

wherein the minimum value is a negative value, and the maximum value is a positive value, and the minimum value is set to -32768, and the maximum value is set to 32767.

13. A non-transitory computer readable media storing instructions that cause a processor to: determine, for a conversion between a current video block of a video and a bitstream of the video, that a differential coding mode used for residual samples is applied to the current video block; and

perform, based on the determining, the conversion using the differential coding mode, wherein, in the differential coding mode, differences between quantized residuals derived with an intra prediction and predictors of the quantized residuals are presented in the bitstream, and wherein at least one of reconstructed quantized residuals is derived based on a sum of one difference and another reconstructed quantized residual,

wherein values of the reconstructed quantized residuals are restricted within a specific range by a clipping operation applied to the reconstructed quantized residuals, and

wherein a single syntax element indicates whether the differential coding mode is applied to two chroma components or not, and the single syntax element is excluded from the bitstream in response to a specific mode being used for the intra prediction of two chroma components.

14. The non-transitory computer readable media of claim 13, wherein each of the values of the reconstructed quantized residuals is restricted by the clipping operation as following:

a value is a minimum value in response to the value being less than the minimum value, and the value is a maximum value in response to the value being greater than the maximum

value, and

wherein the minimum value is a negative value, and the maximum value is a positive value, and the minimum value is set to -32768, and the maximum value is set to 32767.

15. A non-transitory computer readable media storing a bitstream of a video which is generated by a method performed by a video processing apparatus, wherein the method comprises:

determining that a differential coding mode used for residual samples is applied to a current video block of the video; and

generating the bitstream of the video based on the determining using the differential coding mode,

wherein, in the differential coding mode, differences between quantized residuals derived with an intra prediction and predictors of the quantized residuals are presented in the bitstream, and wherein at least one of reconstructed quantized residuals is derived based on a sum of one difference and another reconstructed quantized residual,

wherein values of the reconstructed quantized residuals are restricted within a specific range by applying a clipping operation to the reconstructed quantized residuals, and

wherein a single syntax element indicates whether the differential coding mode is applied to two chroma components or not, and the single syntax element is excluded from the bitstream in response to a specific mode being used for the intra prediction of two chroma components.

16. The non-transitory computer readable media of claim 15, wherein each of the values of the reconstructed quantized residuals is restricted by the clipping operation as following:

a value is a minimum value in response to the value being less than the minimum value, and the value is a maximum value in response to the value being greater than the maximum value, and

wherein the minimum value is a negative value, and the maximum value is a positive value, and the minimum value is set to -32768, and the maximum value is set to 32767.