

CLAIMS

1. A method of video processing, comprising:
 - performing a conversion between a chroma block of a video region of a video picture of a video and a coded representation of the video according to a rule;
 - wherein the rule specifies that, due to the chroma block having a size $M \times N$, the chroma block is disallowed to be represented in the coded representation using an intra coding mode, wherein M and N are integers that indicate a width and a height of the chroma block respectively, and wherein the intra coding mode includes coding the chroma block based on a previously coded video region of the video picture.
2. The method of claim 1, wherein the rule further specifies that a DC prediction mode is used for the chroma block, and wherein the DC prediction mode derives prediction values of a chroma component based on an average of neighboring blocks of the chroma block.
3. The method of claim 1, wherein the rule further specifies that a cross-component linear model (CCLM) prediction mode is used for the chroma block, wherein the CCLM prediction mode uses a linear mode to derive prediction values of a chroma component from another component.
4. The method of claim 2 or 3, wherein $M=2$ chroma samples and $N=8, 16, \text{ or } 32$ chroma samples.
5. The method of claim 2 or 3, wherein the rule further specifies that reconstructed neighboring pixels are not used for the DC prediction mode or the CCLM prediction mode.
6. The method of claim 2, wherein the rule further specifies that neighboring reference pixels are set equal to $1 \ll (\text{bitDepth} - 1)$ for the DC prediction mode.
7. The method of claim 2 or 3, wherein the rule further specifies that a position dependent intra prediction combination (PDPC) method is not applied for the DC prediction mode or the CCLM

prediction mode, and wherein the PDPC method combines neighboring samples with a prediction signal of the video region to generate a refined prediction signal of the video region.

8. The method of claim 2 or 3, wherein the rule further specifies that information about the DC prediction mode or the CCLM prediction mode is derived without being coded for the chroma block.

9. The method of claim 8, wherein the rule further specifies that, for the CCLM prediction mode that satisfies that $\text{predC} = a * \text{recL} + b$, a chroma prediction value, predC , is derived from luma reconstructed value, recL , and parameters, a and b , are initialed to default fixed values at a beginning of the video region.

10. The method of claim 9, wherein a first-in-first-out table is maintained to update the parameters, a and b .

11. The method of any of claims 1 to 10, wherein the rule is applied to another chroma block coded using the CCLM prediction and having a width unequal to M or a height unequal to N .

12. The method of claim 11, wherein CCLM parameters of previously coded video regions are utilized for the video region instead of being derived based on reconstruction information of neighboring blocks of the video region.

13. The method of claim 11, wherein the CCLM parameters are stored in a table and updated using a first-in-first-out table.

14. A method of video processing, comprising:
performing a conversion between a chroma block of a video region of a video picture of a video and a coded representation of the video according to a rule;
wherein the rule specifies that the chroma block having a size $M \times N$ is disallowed to be represented in the coded representation using an intra coding mode in case a dual tree

partitioning is used, wherein M and N are integers that indicate a width and a height of the chroma block, respectively, and

wherein the intra coding mode includes coding the chroma block based on a previously coded video region of the video picture.

15. The method of claim 14, wherein $M=2$ chroma samples and $N=8, 16, \text{ or } 32$ chroma samples.
16. The method of claim 14, wherein the rule further specifies that, due to the dual tree partitioning, a certain partitioning scheme is disabled for the video region that is a coding tree node with a dimension equal to $(M*a) \times (N*b)$, where $M*a$ is a number of chroma samples in a width of the coding tree node and $N*b$ is a number of chroma samples in a height of the coding tree node.
17. The method of claim 16, wherein the certain partitioning scheme is a vertical binary tree (BT) split in which the video region is split into two parts along a vertical direction and wherein $a=2$ and $b=1$.
18. The method of claim 16, wherein the certain partitioning scheme is a horizontal binary tree (BT) split in which the video region is split into two parts along a horizontal direction and wherein $a=1$ and $b=2$.
19. The method of claim 16, wherein the certain partitioning scheme is a vertical ternary tree (TT) split in which the video region is split into three parts along a vertical direction and wherein $a=4$ and $b=1$.
20. The method of claim 16, wherein the certain partitioning scheme is a horizontal ternary tree (TT) split in which the video region is split into three parts along a horizontal direction and wherein $a=1$ and $b=4$.

21. A method of video processing, comprising:
performing a conversion between a chroma block of a video region of a video picture of a video and a coded representation of the video according to a rule;
wherein the rule specifies that the chroma block having a size $M \times N$ is disallowed to be represented in the coded representation using an intra coding mode in case a single tree partitioning is used, wherein M and N are integers that indicate a width and a height of the chroma block, respectively, and
wherein the intra coding mode includes coding the chroma block based on a previously coded video region of the video picture.
22. The method of claim 21, wherein $M=2$ chroma samples and $N=8, 16$, or 32 chroma samples.
23. The method of claim 21 or 22, wherein the rule further specifies that, due to the single tree partitioning, a certain partitioning scheme is disabled for chroma components of the video region that is a coding tree node with a dimension equal to $(M \times a \times \text{subWidthC}) \times (N \times b \times \text{subHeightC})$, where $M \times a$ is a number of luma samples in a width of the coding tree node and $N \times b$ is a number of luma samples in a height of the coding tree node, and subWidthC and subHeightC indicate chroma subsampling ratios in width and height dimensions, respectively.
24. The method of claim 23, wherein the certain partitioning scheme is a vertical binary tree (BT) split in which the video region is split into two parts along a vertical direction and wherein $a=2$ and $b=1$.
25. The method of claim 23, wherein the certain partitioning scheme is a vertical ternary tree (TT) split in which the video region is split into three parts along a vertical direction and wherein $a=4$ and $b=1$.

26. The method of claim 23, wherein the certain partitioning scheme is a horizontal binary tree (BT) split in which the video region is split into two parts along a horizontal direction and wherein $a=1$ and $b=2$.
27. The method of claim 23, wherein the certain partitioning scheme is a horizontal ternary tree (TT) split in which the video region is split into three parts along a horizontal direction and wherein $a=1$ and $b=2$.
28. The method of claim 23, wherein the certain partitioning scheme is a horizontal ternary tree (TT) split in which the video region is split into three parts along a horizontal direction and wherein $a=1$ and $b=4$.
29. The method of claim 23, wherein the chroma block has a dimension equal to $(M*a) \times (N*d)$.
30. The method of claim 23, wherein a coding type of the chroma block is `MODE_TYPE_INTRA` that allows use of an intra mode, a palette mode, and an intra block copy mode for the conversion.
31. The method of claim 23, wherein the rule further specifies that the certain partitioning scheme is allowed for luma components of the coding tree node.
32. The method of claim 31, wherein the coding tree node is split into child nodes and an inter mode and/or an intra block copy mode is allowed for a child node.
33. The method of claim 22, wherein the rule further specifies that a chroma component is split along a collocated luma component and an inter mode and/or an intra block copy mode is allowed for the chroma component.
34. A method of video processing, comprising:

performing a conversion between a chroma block of a video region of a video and a coded representation of the video according to a rule;

wherein the rule specifies that the chroma block having a size $M \times N$ is disallowed to be represented in the coded representation using a combined inter and intra prediction (CIIP) mode that combines an inter prediction with an intra prediction, wherein M and N are integers that indicate a width and a height of the chroma block, respectively.

35. The method of claim 34, wherein the rule further specifies that the CIIP mode is not allowed for chroma components of the video region that is a coding tree node with a dimension equal to $(M \times \text{subWidthC}) \times (N \times \text{subHeightC})$, whereby subWidthC and subHeightC indicate chroma subsampling ratios in width and height dimensions, respectively.

36. The method of claim 34, wherein the performing the conversion includes a chroma blending process that determines a chroma prediction block of the chroma block by blending predictions of the chroma block using the inter prediction in case that the video region is coded using the CIIP mode.

37. The method of claim 36, wherein the rule further specifies that a luma intra prediction is applied to the video region and a chroma intra prediction is not applied to the video region with a dimension equal to $(M \times \text{subWidthC}) \times (N \times \text{subHeightC})$, whereby subWidthC and subHeightC indicate chroma subsampling ratios in width and height dimensions, respectively.

38. The method of claim 34, wherein the rule further specifies that a DC prediction is used for the chroma block having the size $M \times N$ in case that the video region is coded using the CIIP mode.

39. The method of claim 34, wherein the rule further specifies that a CCLM prediction that uses a linear mode to derive prediction values of a chroma component from another component is used for the chroma block having the size $M \times N$ in case that the video region is coded using the CIIP mode.

40. A method of video processing, comprising:
performing a conversion between a video region of a video and a coded representation of the video according to a rule,
wherein the rule specifies that whether and how a maximum transform size applicable to the video region is signaled is dependent on a coding mode type used for representing the video region in the coded representation,
wherein the coding mode type is (i) a near lossless coding mode in which a quantization parameter (QP) applied to the video region is smaller than a threshold, (ii) a lossless coding mode in which a transform of a residual data for the video region is bypassed, or (iii) a lossy coding mode in which the transform of the residual data for the video region is skipped.
41. The method of claim 40, wherein the rule further specifies that the maximum transform size for the lossless coding mode and/or the near lossless coding mode is equal to maxLosslessTbSizeY.
42. The method of claim 40, wherein the lossless coding mode is signaled at a coding unit level using a syntax element, cu_transquant_bypass_flag.
43. The method of claim 40, wherein the threshold is signalled in in a decoder parameter set (DPS), a video parameter set (VPS), a sequence parameter set (SPS), a picture parameter set (PPS), a tile, a slice, or a brick header.
44. The method of claim 40, wherein a transform_skip_flag is equal to 1 for the near lossless coding mode of the video region.
45. The method of claim 40, wherein the rule further specifies that a first maximum transform size for the lossless coding mode and/or the near lossless coding mode of the video region is set based on i) another maximum transform size used for the lossy coding mode, ii) a maximum coding tree unit size, and/or iii) a maximum transform skip size.

46. The method of claim 40, wherein the maximum transform size for the lossless coding mode or the near lossless coding mode of the video region is signalled at a video unit level corresponding to a syntax element (SE) at a sequence parameter set (SPS), a video parameter set (VPS), a picture parameter set (PPS), a sub-picture, a slice, a tile, a brick, a coding tree unit (CTU), or a virtual pipeline data unit (VPDU).
47. The method of claim 46, wherein the SE is signalled depending on a maximum coding tree block (CTB) size, the maximum transform size, and/or a maximum transform skip size.
48. The method of claim 47, wherein the SE is signalled in case that $\log_2_ctu_size_minus5$ is larger than N, whereby N is an integer.
49. The method of claim 47, wherein the SE is signalled in case that $sps_max_luma_transform_size_64_flag$ is equal to M, whereby M is an integer.
50. The method of claim 47, wherein the SE is signalled in case that $\log_2_transform_skip_max_size_minus2$ is larger than K.
51. The method of claim 47, wherein a difference between $\log(maxLosslessTbSizeY)$ and X is signalled, and wherein $maxLosslessTbSizeY$ indicates the maximum transform size for the near lossless coding mode or the lossless coding mode of the video region.
52. The method of claim 51, wherein X is equal to $\log_2_ctu_size_minus5$, $sps_max_luma_transform_size_64_flag$, $\log_2_transform_skip_max_size_minus2$, or a fixed integer.
53. The method of claim 47, wherein a difference between $maxLosslessTbSizeY$ and X is signalled, and wherein $maxLosslessTbSizeY$ indicates the maximum transform size for the near lossless coding mode or the lossless coding mode of the video region.

54. The method of claim 47, wherein a value corresponding to the difference between $\text{maxLosslessTbSizeY}$ and $X - K$ is signalled, wherein $\text{maxLosslessTbSizeY}$ indicates the maximum transform size for the near lossless coding mode or the lossless coding mode of the video region and K is an integer.
55. The method of claim 53 or 54, wherein X is equal to CtbSizeY , MaxTbSize , MaxTsSize , or a fixed integer.
56. The method of claim 46, wherein $\text{maxLosslessTbSizeY}$ of the video region is derived from the SE, and wherein $\text{maxLosslessTbSizeY}$ indicates the maximum transform size for the near lossless coding mode or the lossless coding mode of the video region.
57. The method of claim 46, wherein the syntax element corresponds to a flag.
58. The method of claim 57, wherein in case that the flag is not present, a value that is a positive integer is inferred.
59. The method of claim 40, wherein the rule further specifies that the maximum transform size is derived for the lossless coding mode and/or the near lossless coding mode of the video region.
60. The method of claim 59, wherein the maximum transform size is set equal to i) a fixed integer value, ii) another maximum transform size used for the lossy coding mode, iii) a maximum coding tree unit size, iv) a VPDU size, v) a maximum transform skip block size, or vi) a maximum residue coding block size.
61. A method of video processing, comprising:
performing a conversion between a video region of a video and a coded representation of the video according to a rule,

wherein the rule specifies that a residual block of a video region is divided into one or more residual sub-blocks for a lossless coding and/or a near lossless coding based on a maximum transform size of the video region to which the lossless coding and/or the near lossless coding is applied, and

wherein the lossless coding includes bypassing a transform of the residual block for the video region and the near lossless coding uses a quantization parameter (QP) that is smaller than a threshold.

62. The method of claim 61, wherein the rule further specifies that a width and a height of the one or more residual sub-blocks obtained after the division are not greater than the maximum transform size.

63. The method of claim 62, wherein the rule further specifies to repeat dividing of the residual block until the width and the height of the one or more residual sub-blocks are not greater than the maximum transform size.

64. The method of any of claims 1 to 63, wherein the conversion includes encoding the video into the coded representation.

65. The method of any of claims 1 to 63, wherein the conversion includes decoding the coded representation to generate the video.

66. A video processing apparatus comprising a processor configured to implement a method recited in any one or more of claims 1 to 65.

67. A computer readable medium storing program code that, when executed, causes a processor to implement a method recited in any one or more of claims 1 to 65.

68. A computer readable medium that stores a coded representation or a bitstream representation generated according to any of the above described methods.