

CLAIMS

What is claimed is:

1. A method of video processing, comprising:
determining, for a conversion between a current video unit of a video comprising one or more video blocks and a bitstream representation of the video, a padding process used for padding unavailable samples during application of a cross-component adaptive loop filtering (CC-ALF) tool to at least some video blocks of the current video unit according to a rule; and
performing the conversion based on the determining;
wherein the rule specifies that the padding process is also used for padding unavailable samples during application of an adaptive loop filtering (ALF) tool to one or more video blocks of the current video unit.
2. The method of claim 1, wherein the ALF tool is applied to a color component of the current video unit and the ALF tool includes correcting samples values of the color component using an information provided by the color component of the current video unit.
3. The method of any one or more of claims 1 to 2, wherein the CC-ALF tool is applied to a first color component of the current video unit and includes correcting sample values of a first component of the current video unit using sample values of a second component of the current video unit.
4. The method of any one or more of claims 1 to 3, wherein the current video unit corresponds to a slice, a tile, a brick, a sub-picture, a picture, or a 360-degree virtual picture bounded by 360-degree virtual boundaries.
5. The method of any one or more of claims 1 to 4, wherein the padding process corresponds to mirrored padding that is applied on the unavailable samples at one or more ALF virtual boundaries associated with the usage of the CC-ALF tool.
6. The method of any one or more of claims 1 to 4, wherein the padding process corresponds to repetitive padding that is applied on the unavailable samples at one of: a slice, a

tile, a brick, a picture, a sub-picture, or a 360-degree virtual boundary associated with the usage of the CC-ALF.

7. The method of any one or more of claims 1 to 4, wherein the padding process corresponds to repetitive padding is applied on one or more boundaries except at the one or more ALF virtual boundaries associated with the usage of the CC-ALF.

8. The method of claim 7, wherein the one or more boundaries include one of: all boundaries, horizontal boundaries, or vertical boundaries.

9. The method of any one or more of claims 1 to 4, wherein the padding process corresponds to mirrored padding is applied on one or more boundaries except at the one or more ALF virtual boundaries associated with the usage of the CC-ALF.

10. The method of claim 9, wherein the one or more boundaries include one of: all boundaries, horizontal boundaries, or vertical boundaries.

11. The method of any one or more of claims 1 to 10, wherein the current video unit corresponds to an ALF processing unit, a coding tree unit (CTU), or a basic ALF processing unit.

12. The method of claim 11, wherein the ALF processing unit defines a unit bounded by two horizontal boundaries and two vertical boundaries.

13. The method of claim 11, wherein the two horizontal boundaries include one of: two ALF virtual boundaries, or one ALF virtual boundary and one boundary of a picture/slice /tile /brick /sub-picture or a 360-degree virtual boundary.

14. The method of claim 11, wherein the basic ALF processing unit is generated based on recursively splitting an ALF processing unit, a CTU, or a narrow ALF processing unit, and wherein boundaries of the basic ALF processing unit are not crossed by brick/slice/tile/sub-picture boundary, and/or 360-degree virtual boundary, and/or an ALF virtual boundary where filtering across a boundary is disallowed.

15. The method of any one or more of claims 1 to 4, wherein the padding process corresponds to repetitive padding that is based on whether a neighboring sample of the current video unit is in a different neighboring slice.

16. The method of claim 15, wherein a current slice of the current video unit is a raster-scan slice and wherein the neighboring slice of the current slice is also a raster-scan slice.

17. The method of any one or more of claims 15 to 16, wherein the current video unit is an ALF processing unit, and wherein an unavailable sample is padded using its nearest sample in the ALF processing unit.

18. A method of video processing, comprising:
performing a conversion between a video unit of a video and a bitstream representation of the video,

wherein, during the conversion, unavailable samples of the video unit are padded in a predefined padding order according to a rule in an application of an adaptive loop filtering (ALF) process or a cross-component adaptive loop filtering (CC-ALF) process.

19. The method of claim 18, wherein the video unit corresponds to a processing unit and the rule specifies the predefined padding order for padding the unavailable samples of the video unit.

20. The method of claim 18, wherein the predefined padding order includes padding the unavailable samples using at least one row or at least one column of the video unit, and wherein a location of the at least one row or the at least one column is determined based on locations of the unavailable samples.

21. The method of claim 18, wherein the predefined padding order includes padding the unavailable samples using at least one sample of the video unit, and wherein a location of the at least one sample is determined based on locations of the unavailable samples.

22. The method of claim 18, wherein an availability of at least one neighboring sample of the video unit is determined based on a relationship between the one or more

neighboring samples and a parameter (ctuDown or ctuUp) associated with the video region, and wherein the video region corresponds to a processing unit that includes the ctuUP corresponding to N rows of a coding tree unit (CTU) and the ctuDown corresponding to 'CtbSize - M' rows of the CTU, whereby M is an integer.

23. The method of claim 22, wherein, if a neighboring sample is in a second video unit that is different from the ctuDown, then the rule specifies that filtering across the second video unit is disallowed.

24. The method of claim 18, wherein the video unit corresponds to a current coding tree unit included in a raster-scan slice and the rule specifies the predefined padding order for padding the unavailable samples of the video unit.

25. The method of claim 18, wherein the predefined padding order includes padding the unavailable samples located in an above-left CTU that is in a slice different from the raster-scan slice using its nearest neighboring sample located in an above CTU that is in the raster-scan slice.

26. The method of claim 18, wherein the predefined padding order includes padding the unavailable samples located in an above-left CTU that is in a slice different from the raster-scan slice using its nearest neighboring sample located in a left CTU that is in the raster-scan slice.

27. The method of claim 18, wherein the predefined padding order includes padding the unavailable samples located in an below-right CTU that is in a slice different from the raster-scan slice using its nearest neighboring sample located in a below CTU that is in the raster-scan slice.

28. The method of claim 18, wherein the predefined padding order includes padding the unavailable samples located in an below-right CTU that is in a slice different from the raster-scan slice using its nearest neighboring sample located in a right CTU that is in the raster-scan slice.

29. A method of video processing, comprising:
determining, for a video region of a video for which an application of an adaptive loop filter (ALF) is enabled, that the video region is crossed by a boundary of a video unit; and
performing a conversion between the video and a bitstream representation of the video, wherein, for the conversion, the video region is split into multiple partitions according to a rule due to the video region being crossed by the boundary of the video unit.
30. The method of claim 29, wherein a filtering process across the boundary is disallowed.
31. The method of claim 29, wherein the video region corresponds to an ALF processing unit, a narrow ALF processing unit, or a coding tree unit.
32. The method of claim 29, wherein the video unit corresponds to a brick, a slice, a tile, a sub-picture, or a 360-degree virtual picture.
33. The method of claim 29, wherein the rule further specifies to split the video region recursively until the video region is not across the boundary of the video unit.
34. The method of claim 29 or 30, wherein the filtering process corresponds to an ALF process or a CC-ALF process and wherein the rule further specifies that the filtering process is performed after a splitting process to split the video region is finished.
35. The method of claim 29, wherein a padding process used for padding at least one sample is performed on a basic ALF processing unit that is obtained after splitting the video region until the video region is not across the boundary of the video unit.
36. The method of claim 29, wherein the rule specifies to check different horizontal boundaries in a predefined order to generate a basic ALF processing unit that is obtained after splitting the video region until the video region is not across the boundary of the video unit.

37. The method of claim 36, wherein the predefined order is an ALF virtual boundary, a 360-degree virtual boundary, and a coding tree unit (CTU) boundary that is a boundary of the video unit.

38. The method of claim 36, wherein the predefined order is a 360-degree virtual boundary, an ALF virtual boundary, and a coding tree unit (CTU) boundary that is a boundary of the video unit.

39. The method of claim 36, wherein the rule specifies different predefined orders for a top boundary and a bottom boundary of the video unit.

40. The method of claim 29, wherein the rule specifies to check different vertical boundaries in a predefined order to generate a basic ALF processing unit that is obtained after splitting the video region until the video region is not across the boundary of the video unit.

41. The method of claim 40, wherein the predefined order is a 360-degree virtual boundary and a coding tree unit (CTU) boundary that is a boundary of the video unit.

42. The method of claim 40, wherein the predefined order is a coding tree unit (CTU) boundary and a 360-degree virtual boundary of the video unit.

43. The method of claim 40, wherein the rule specifies different predefined orders for a left boundary and a right boundary of the video unit.

44. A method of video processing, comprising:
performing a conversion between a video unit of a video and a bitstream representation of the video according to a rule,
wherein the rule specifies that applying an adaptive loop filtering (ALF) and/or a cross-component adaptive loop filtering (CC-ALF) to a sample located at a boundary of the video unit is disallowed in case that a filtering process across the boundary is disallowed.

45. The method of claim 44, wherein applying the ALF and/or the CC-ALF is disabled upon determining that at least N neighboring samples associated with the ALF and/or the CC-ALF are unavailable, whereby N is an integer.

46. The method of claim 44, wherein the boundary of the video unit is a horizontal boundary, and wherein applying the ALF and/or the CC-ALF is disabled upon determining that a vertical distance between the sample and the horizontal boundary of the video unit is smaller than or equal to M samples, whereby M is an integer.

47. The method of claim 44, wherein the boundary of the video unit is a vertical boundary, and wherein applying the ALF and/or the CC-ALF is disabled upon determining that a horizontal distance between the sample and the vertical boundary of the video unit is smaller than or equal to M samples, whereby M is an integer.

48. A method of video processing, comprising:
performing a conversion between a video unit of a video and a bitstream representation of the video,
wherein the bitstream representation conforms to a format rule,
wherein the video region is different from a coding tree block, and
wherein the format rule specifies whether a syntax element is included in the bitstream representation indicative of an applicability of an adaptive loop filtering (ALF) tool and/or a cross-component adaptive loop filtering (CC-ALF) tool to the video region.

49. The method of claim 48, wherein the video region corresponds to an ALF processing unit.

50. The method of claim 48, wherein the applicability of the ALF tool and/or the CC-ALF tool is explicitly indicated in the bitstream representation.

51. The method of claim 48, wherein the applicability of the ALF tool and/or the CC-ALF tool is derived.

52. A method of video processing, comprising:

determining, for a conversion between a video unit of a video and a bitstream representation of the video, an applicability of a cross-component adaptive loop filtering (CC-ALF) tool to samples of the video unit according to a rule; and
performing the conversion according to the determining;
wherein the bitstream representation includes an indication that the CC-ALF is available for the video unit, and
wherein the rule specifies one or more conditions that override the indication.

53. The method of claim 52, wherein the one or more conditions include a spatial position of the sample with respect to the video unit.

54. The method of claim 52, wherein the one or more conditions include on luma sample values used in the CC-ALF.

55. The method of claim 52, wherein the one or more conditions include differences in sample values between the sample and its neighboring samples.

56. The method of claim 52, wherein the one or more conditions include a relationship between a value of an offset derived using filtered luma samples corresponding to a chroma sample and a threshold value.

57. The method of claim 52, wherein the one or more conditions include coded information of a block including a chroma sample and/or neighboring blocks of the block.

58. A method of video processing, comprising:
performing a conversion between a video unit of a video and a bitstream representation of the video according to a rule,
wherein the rule specifies that an arithmetic used during the conversion omits at least one of three clipping operations that include a first clipping operation corresponding to a chroma adaptive loop filtering (ALF) filtering, a second clipping operation corresponding to a cross-component adaptive loop filtering (CC-ALF) offset derivation, and a third clipping operation corresponding to a refinement of a chroma filtered sample to derive a final chroma sample value.

59. The method of claim 58, wherein the rule specifies omitting the first clipping operation in case that applying the CC-ALF is enabled.

60. The method of claim 58, wherein the rule specifies omitting the second clipping operation in case that applying the CC-ALF is disabled.

61. The method of claim 58, wherein the rule specifies omitting the first clipping operation and the second clipping operation.

62. A method of video processing, comprising:
making a determination, for a conversion between a first video unit of a video and a bitstream representation of the video, of a cross-component adaptive loop filtering (CC-ALF) offset according to a rule; and
performing the conversion based on the determination, and
wherein the rule specifies that the CC-ALF offset is clipped to a first range different from a second range that is expressed as $[-(1 \ll (\text{BitDepthC} - 1)), (1 \ll (\text{BitDepthC} - 1)) - 1]$, wherein BitDepthC is a bit depth value.

63. The method of claim 62, wherein the first range lacks a dependency on the bit depth value.

64. The method of claim 62, wherein a third range associated with a second video unit is defined differently from the first range associated with the first video unit.

65. The method of claim 62, wherein the first range is derived from a non-linear clipping range used in another clipping operation corresponding to a chroma ALF filtering.

66. A method of video processing, comprising:
deriving, for a conversion between a first video unit of a video and a bitstream representation of the video, a cross-component adaptive loop filtering (CC-ALF) offset according to a rule; and
performing the conversion using the CC-ALF offset, and

wherein the rule specifies that the CC-ALF offset is rounded with a rounding offset based on a bit-depth value instead of a fixed value.

67. The method of claim 66, wherein the rounding offset depends on bit-depth difference between a first color component and a second color component.

68. The method of claim 66, wherein the CC-ALF offset is rounded using a following expression: $\text{sum} = \text{alfPicture}[x\text{CtbC} + x][y\text{CtbC} + y] + (\text{sum} + \text{Off}) \gg (7 + (\text{BitDepthY} - \text{BitDepthC}))$ and $\text{ccAlfPicture}[x\text{CtbC} + x][y\text{CtbC} + y] = \text{Clip3}(0, (1 \ll (\text{BitDepthC} - 1), \text{sum})$, wherein *Off* is set equal to $(1 \ll (6 + (\text{BitDepthY} - \text{BitDepthC})))$, wherein $\text{alfPicture}[x\text{CtbC} + x][y\text{CtbC} + y]$ represents a chroma sample after applying the first clipping operation, wherein *sum* represents a summation of CC-ALF filter coefficients times luma samples in a CC-ALF filter support after applying the second clipping operation, wherein *ccAlfPicture* represents the final chroma sample value after applying the third clipping operation, and wherein *BitDepthC* and *BitDepthY* represent bit-depths of luma and chroma components, respectively.

69. A method of video processing, comprising:
performing a conversion between a video unit of a video and a bitstream representation of the video according to a rule,
wherein the rule specifies that one or more processing steps used during a chroma adaptive loop filtering (ALF) process and/or a cross-component adaptive loop filtering (CC-ALF) process applied to samples of the video unit are same.

70. The method of claim 69, wherein the conversion includes a clipping operation for clipping a chroma sample generated by the chroma ALF process and/or the CC-ALF process that results in the chroma sample to lie within a predefined range.

71. The method of claim 70, wherein the predefined range is based at least in part on a chroma bit depth value.

72. The method of claim 70, wherein the clipping operation is applied only during the CC-ALF process if the CC-ALF process is applied after the chroma ALF process.

73. The method of claim 70, wherein the clipping operation is applied only during the chroma ALF process if the CC-ALF process is applied before the chroma ALF process.

74. A method of video processing, comprising:
performing a conversion between a video unit of a video and a bitstream representation of the video according to a rule,
wherein the rule specifies that, during the conversion, a value of a sample of a first color component of the video unit is modified by applying a modification using an information of a second color component of the video unit, wherein the modification is based on one or more parameters used in an adaptive loop filtering (ALF) process for the video unit.

75. The method of claim 74, wherein the one or more parameters used in an adaptive loop filtering process are clipping parameters, wherein the clipping parameters used on a first sample value are different from clipping parameters used on a second sample value, wherein the first sample value and the second sample value are included in the video unit.

76. The method of claim 75, wherein the clipping parameters are defined to lie in a range having a minimum clipping value and a maximum clipping value.

77. The method of claim 76, wherein the minimum clipping value and/or the maximum clipping value are explicitly signaled in the bitstream representation.

78. The method of claim 76, wherein the minimum clipping value and/or the maximum clipping value are based in part on the value of the sample.

79. The method of claim 76, wherein the minimum clipping value and/or the maximum clipping value are based in part on a spatial location of the value of the sample with respect to the video unit.

80. The method of claim 76, wherein the minimum clipping value and/or the maximum clipping value are signaled or predefined.

81. The method of claim 76, wherein the minimum clipping value and/or the maximum clipping value are derived from other parameters.

82. A method of video processing, comprising:
making a determination, for a conversion between a first sub-picture of a video and a bitstream representation of the video, whether a cross-component loop filtering (CC-ALF) is applicable to a sample of the first sub-picture based on a rule; and
performing the conversion based on the determining;
wherein the CC-ALF for the sample uses samples from a second sub-picture; and
wherein the rule is based on whether loop filtering across a sub-picture boundary is allowed for the first sub-picture and/or the second sub-picture.

83. The method of claim 82, wherein the CC-ALF is skipped on the sample in case that the loop filtering across the sub-picture boundary is disallowed for the first sub-picture.

84. The method of claim 82, wherein the CC-ALF is skipped on the first sample in case that the loop filtering across the sub-picture boundary is disallowed for the second sub-picture.

85. The method of claim 82, wherein the CC-ALF is skipped on the first sample in case that the loop filtering across the sub-picture boundary is disallowed for the first sub-picture and/or the second sub-picture.

86. The method of one or more of claims 82 to 84, wherein whether the loop filtering across the sub-picture boundary is allowed is based on a flag associated with the first sub-picture and/or the second sub-picture.

87. The method of any of previous claims, wherein the method is further based on at least one of: 1) a type of video contents, 2) a message signaled in a sequence parameter set (SPS), a video parameter set (VPS), a picture parameter set (PPS), a dependency parameter set (DPS), an adaptation parameter set (APS), a picture header, a slice header, a tile group header, a largest coding unit (LCU), a coding unit (CU), a LCU row, a group of LCUs, a transform unit (TU), a prediction unit (PU) block, or a video coding unit, 3) a position of CU, PU, TU, block, or video coding unit, 4) decoded information of a current block and/or a neighboring block, 5) a dimension or shape of the current block and/or the neighboring block, 6) an indication of a color format, 7) a coding tree structure, 8) a slice type, a tile group type, and/or a picture type, 9) a type of a color component, 10) a temporal layer identifier, 11) profiles, levels, or tiers of a standard.

88. The method of any of claims 1 to 87, wherein the conversion includes encoding the video into the bitstream representation.

89. The method of any of claims 1 to 87, wherein the conversion includes decoding the video from the bitstream representation.

90. A video processing apparatus comprising a processor configured to implement a method recited in any one or more of claims 1 to 89.

91. 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 89.

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