CLAIMS

1. A method of filtering a sample value of a picture,

wherein the method comprises:

obtaining a reconstructed sample value for a block of the picture;

obtaining filter coefficients for adaptive loop filtering according to a bitstream;

obtaining a sum value according to the filter coefficients and the reconstructed sample value for the block;

rounding the sum value according to a vertical position of a sample in the block, to obtain a rounded sum value, wherein the vertical position of the sample in the block is a luma sample position, and wherein the rounded sum value equals curr + ((sum + roundY) >> alfShiftY) and satisfies the following table:

Condition	**roundY**	**alfShiftY**	**y1**	**y2**	**y3**
(y = = CtbSizeY − 5 || y = = CtbSizeY − 4) && (applyAlfLineBufBoundary = = 1)	512	10	0	0	0
(y = = CtbSizeY − 6 || y = = CtbSizeY − 3) && (applyAlfLineBufBoundary = = 1)	64	7	1	1	1
(y = = CtbSizeY − 7 || y = = CtbSizeY − 2) && (applyAlfLineBufBoundary = = 1)	64	7	1	2	2
Otherwise	64	7	1	2	3

wherein

y represents the vertical position of the sample;

curr represents the reconstructed sample value for the block;

sum represents the sum value;

CtbSizeY represents a size of a coding tree block (CTB) corresponding to the block;

y1, y2 and y3 represent vertical sample position offsets indicating other reconstructed sample values involved in the obtaining the sum value;

roundY and alfShiftY are variables that are determined based on the vertical position of the sample;

apply AlfLineBufBoundary is an index and derived as follows:

if a bottom boundary of the coding tree block is a bottom boundary of the picture and pic_height_in_luma_samples−yCtb≤CtbSizeY−4,

apply AlfLineBufBoundary is set equal to 0, otherwise,

apply AlfLineBufBoundary is set equal to 1,

wherein the pic_height_in_luma_samples represents a height of the picture in luma samples, and (xCtb, yCtb) represents a location of the CTB; and

obtaining a filter reconstructed sample value for the block according to the rounded sum value.

2. The method of claim 1, wherein alfShiftY is equal to 10 when the vertical position of the sample is satisfied any one of the below conditions:

y = = CtbSizeY – t-1, or

y = = CtbSizeY – t, or

y = = ctbHeightC – t, or

y = = ctbHeightC – t-1,

wherein y is the vertical position of the sample, CtbSizeY is a coding tree block, CTB, size, ctbHeightC is the height of the CTB, t is an integer value.

3. The method of claim 1 or 2, wherein alfShiftY is equal to 7 when the vertical position of the sample is not satisfied any one of the below conditions:

y = = CtbSizeY – t-1, or

y = = CtbSizeY – t, or

y = = ctbHeightC – t, or

y = = ctbHeightC – t-1,

wherein y is the vertical position of the sample, CtbSizeY is a coding tree block, CTB, size, ctbHeightC is the height of the CTB, t is an integer value.

4. The method of any one of claims 1 to 3, wherein when y is equal to 0, alfShiftY is equal to 10.

5. The method of any one of claims 1 to 4, wherein when y is not equal to 0, alfShiftY is equal to 7.

6. The method of any one of claims 1 to 5, wherein the rounding process comprises:

adding a value which is determined based on the vertical position to the sum value in order to obtain an added sum value, and

shifting the added sum value based on the vertical position.

7. A video decoding apparatus, comprising:

a reconstructing module, which is configured to obtain a reconstructed sample value for a block of a picture;

a parsing module, which is configured to obtain filter coefficients for adaptive loop filtering according to a bitstream;

a calculating module, which is configured to obtain a sum value according to the filter coefficients and the reconstructed sample value for the block;

a rounding module, which is configured to round the sum value according to a vertical position of a sample in the block, to obtain a rounded sum value, wherein the vertical position of the sample in the block is a luma sample position, and wherein the rounded sum value equals curr + ((sum + roundY) >> alfShiftY) and satisfies the following table:

Condition	**roundY**	**alfShiftY**	**y1**	**y2**	**y3**
(y = = CtbSizeY − 5 || y = = CtbSizeY − 4) && (applyAlfLineBufBoundary = = 1)	512	10	0	0	0
(y = = CtbSizeY − 6 || y = = CtbSizeY − 3) && (applyAlfLineBufBoundary = = 1)	64	7	1	1	1
(y = = CtbSizeY − 7 || y = = CtbSizeY − 2) && (applyAlfLineBufBoundary = = 1)	64	7	1	2	2
Otherwise	64	7	1	2	3

wherein

y represents the vertical position of the sample;

curr represents the reconstructed sample value for the block;

sum represents the sum value;

CtbSizeY represents a size of a coding tree block (CTB) corresponding to the block;

y1, y2 and y3 represent vertical sample position offsets indicating other reconstructed sample values involved in the obtaining the sum value;

roundY and alfShiftY are variables that are determined based on the vertical position of the sample;

apply AlfLineBufBoundary is an index and derived as follows:

if a bottom boundary of the coding tree block is a bottom boundary of the picture and pic_height_in_luma_samples−yCtb≤CtbSizeY−4,

apply AlfLineBufBoundary is set equal to 0, otherwise,

apply AlfLineBufBoundary is set equal to 1,

wherein the pic_height_in_luma_samples represents a height of the picture in luma samples, and (xCtb, yCtb) represents a location of the CTB; and

a filtering module, which is configured to obtain a filter reconstructed sample value for the block according to the rounded sum value.

8. The video decoding apparatus of claim 7, wherein alfShiftY is equal to 10 when the vertical position of the sample is satisfied any one of the below conditions:

y = = CtbSizeY – t-1, or

y = = CtbSizeY – t, or

y = = ctbHeightC – t, or

y = = ctbHeightC – t-1,

wherein y is the vertical position of the sample, CtbSizeY is a coding tree block, CTB, size, ctbHeightC is the height of the CTB, t is an integer value.

9. The video decoding apparatus of claim 7 or 8, wherein alfShiftY is equal to 7 when the vertical position of the sample is not satisfied any one of the below conditions:

y = = CtbSizeY – t-1, or

y = = CtbSizeY – t, or

y = = ctbHeightC – t, or

y = = ctbHeightC – t-1,

wherein y is the vertical position of the sample, CtbSizeY is a coding tree block, CTB, size, ctbHeightC is the height of the CTB, t is an integer value.

10. The video decoding apparatus of any one of claims 7 to 9, wherein when y is equal to 0, alfShiftY is equal to 10.

11. The video decoding apparatus of any one of claims 7 to 10, wherein when y is not equal to 0, alfShiftY is equal to 7.

12. The video decoding apparatus of any one of claims 7 to 11, wherein the rounding module is configured to

add a value which is determined based on the vertical position to the sum value in order to obtain an added sum value, and shift the added sum value based on the vertical position, to obtain the rounded sum value.

13. A decoder, comprising:

one or more processors; and

a non-transitory computer-readable storage medium coupled to the processors and storing programming for execution by the processors, wherein the programming, when executed by the processors, configures the decoder to carry out the method according to any one of the claims 1 to 6.

14. A non-transitory computer-readable medium comprising an encoded bitstream for a video signal decoded by performing the method of any one of the claims 1 to 6.