(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
22 October 2020 (22.10.2020)

WIPO | PCT

(10) International Publication Number
WO 2020/211770 A1

(51) International Patent Classification:
H04N 19/82 (2014.01)

(21) International Application Number:
PCT/CN2020/084876

(22) International Filing Date:
15 April 2020 (15.04.2020)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
PCT/CN2019/082626
15 April 2019 (15.04.2019) CN

(71) Applicants: **BEIJING BYTEDANCE NETWORK TECHNOLOGY CO., LTD.** [CN/CN]; Room B-0035, 2/F, No.3 Building, No.30, Shixing Road, Shijingshan District, Beijing 100041 (CN). **BYTEDANCE INC.** [US/US]; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US).

(72) Inventors: **ZHANG, Li**; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US). **ZHANG, Kai**; 12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137, Los Angeles, California 90066 (US). **LIU, Hongbin**; Jinritoutiao Post Office, China Satellite Communications Tower, No.63, Zhichun Road, Haidian District, Beijing 100080 (CN). **WANG, Yue**; Jinritoutiao Post Office, China Satellite Communications Tower, No.63, Zhichun Road, Haidian District, Beijing 100080 (CN).

(74) Agent: **LIU, SHEN & ASSOCIATES**; 10th Floor, Building 1, 10 Caihefang Road, Haidian District, Beijing 100080 (CN).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ,

(54) Title: TEMPORAL PREDICTION OF PARAMETERS IN NON-LINEAR ADAPTIVE LOOP FILTER

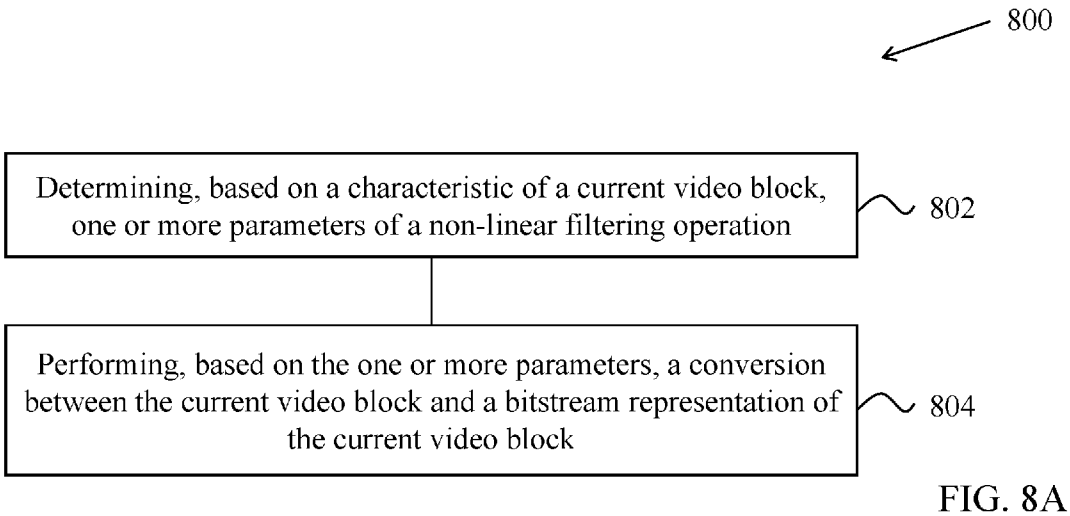

FIG. 8A

(57) Abstract: Devices, systems and methods for temporal prediction of parameters in non-linear adaptive loop filtering are described. In an exemplary aspect, a method for visual media processing includes configuring, for a current video block, one or more parameters of a clipping operation that is part of a non-linear filtering operation; and performing, based on the one or more parameters, a conversion between the current video block and a bitstream representation of the current video block, wherein the one or more parameters is coded in accordance with a rule.

WO 2020/211770 A1

[Continued on next page]

TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *of inventorship (Rule 4.17(iv))*

Published:

— *with international search report (Art. 21(3))*

TEMPORAL PREDICTION OF PARAMETERS IN NON-LINEAR ADAPTIVE LOOP FILTER

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] Under the applicable patent law and/or rules pursuant to the Paris Convention, this application is made to timely claim the priority to and benefit of International Patent Application No. PCT/CN2019/082626, filed on April 15, 2019. For all purposes under the law, the entire disclosure of the aforementioned application is incorporated by reference as part of the disclosure of this application.

TECHNICAL FIELD

[0002] This patent document relates to video coding and decoding techniques, devices and systems.

BACKGROUND

[0003] 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

[0004] Devices, systems and methods related to digital video coding, and specifically, to temporal prediction of parameters in non-linear adaptive loop filtering are described. The described methods may be applied to both the existing video coding standards (e.g., High Efficiency Video Coding (HEVC)) and future video coding standards (e.g., Versatile Video Coding (VVC)) or codecs.

[0005] In one representative aspect, the disclosed technology may be used to provide a method for visual media processing. This method includes configuring, for a current video block, one or more parameters of a clipping operation that is part of a non-linear filtering operation; and performing, based on the one or more parameters, a conversion between the current video block and a bitstream representation of the current video block, wherein the one or more parameters is coded in accordance with a rule.

[0006] In another representative aspect, the disclosed technology may be used to provide a method for visual media processing. This method includes determining, based on a characteristic of a current video block, one or more parameters of a non-linear filtering

operation; and performing, based on the one or more parameters, a conversion between the current video block and a bitstream representation of the current video block.

[0007] In yet another representative aspect, the disclosed technology may be used to provide a method for visual media processing. This method includes configuring, for a current video block, one or more parameters of a clipping operation that is part of a non-linear filtering operation; and performing, based on the one or more parameters, a conversion between the current video block and a bitstream representation of the current video block, wherein the one or more parameters are presented in the bitstream representation independently from values of at least one filter coefficient associated with the non-linear filtering operation.

[0008] In yet another representative aspect, the disclosed technology may be used to provide a method for visual media processing. This method includes configuring, for a current video block, one or more parameters of a clipping operation that is part of a non-linear filtering operation; and performing, based on the one or more parameters, a conversion between the current video block and a bitstream representation of the current video block, wherein the current video block inherits filter coefficients from an i-th filter, and wherein a first rule associated with inheritance of the one or more parameters of the clipping operation is different from a second rule associated with inheritance of the filter coefficients.

[0009] In yet another representative aspect, the above-described method is embodied in the form of processor-executable code and stored in a computer-readable program medium.

[0010] In yet another representative aspect, a video encoder apparatus may implement a method as described herein.

[0011] In yet another representative aspect, a video decoder apparatus may implement a method as described herein.

[0012] The above and other aspects and features of the disclosed technology are described in greater detail in the drawings, the description and the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] FIG. 1 shows an example of an encoder block diagram for video coding.

[0014] FIGS. 2A, 2B and 2C show examples of geometry transformation-based adaptive loop filter (GALF) filter shapes.

[0015] FIG. 3 shows an example of a flow graph for a GALF encoder decision.

[0016] FIGS. 4A−4D show example subsampled Laplacian calculations for adaptive loop filter (ALF) classification.

[0017] FIG. 5 shows an example of neighboring samples utilized in a bilateral filter.

[0018] FIG. 6 shows an example of windows covering two samples utilized in a weight calculation.

[0019] FIG. 7 shows an example of a scan pattern.

[0020] FIGS. 8A–8C show flowcharts of exemplary methods for the temporal prediction of parameters in non-linear adaptive loop filtering.

[0021] FIG. 9 is a block diagram of an example of a hardware platform for implementing a video decoding or a video encoding technique described in the present document.

[0022] FIG. 10 is a block diagram of an example video processing system in which disclosed techniques may be implemented.

[0023] FIG. 11 shows a flowchart of an example method for visual media processing.

[0024] FIG. 12 shows a flowchart of an example method for visual media processing.

[0025] FIG. 13 shows a flowchart of an example method for visual media processing.

[0026] FIG. 14 shows a flowchart of an example method for visual media processing.

DETAILED DESCRIPTION

[0027] Due to the increasing demand of higher resolution video, video coding methods and techniques are ubiquitous in modern technology. Video codecs typically include an electronic circuit or software that compresses or decompresses digital video, and are continually being improved to provide higher coding efficiency. A video codec converts uncompressed video to a compressed format or vice versa. There are complex relationships between the video quality, the amount of data used to represent the video (determined by the bit rate), the complexity of the encoding and decoding algorithms, sensitivity to data losses and errors, ease of editing, random access, and end-to-end delay (latency). The compressed format usually conforms to a standard video compression specification, e.g., the High Efficiency Video Coding (HEVC) standard (also known as H.265 or MPEG-H Part 2), the Versatile Video Coding (VVC) standard to be finalized, or other current and/or future video coding standards.

[0028] 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 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). 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.

[0029] Embodiments of the disclosed technology may be applied to existing video coding standards (e.g., HEVC, H.265) and future standards to improve runtime performance. Section headings are used in the present document to improve readability of the description and do not in any way limit the discussion or the embodiments (and/or implementations) to the respective sections only.

1 Examples of color space and chroma subsampling

[0030] Color space, also known as the color model (or color system), is an abstract mathematical model which simply describes the range of colors as tuples of numbers, typically as 3 or 4 values or color components (e.g. RGB). Basically speaking, color space is an elaboration of the coordinate system and sub-space.

[0031] For video compression, the most frequently used color spaces are YCbCr and RGB.

[0032] YCbCr, Y'CbCr, or Y Pb/Cb Pr/Cr, also written as YCBCR or Y'CBCR, is a family of color spaces used as a part of the color image pipeline in video and digital photography systems. Y′ is the luma component and CB and CR are the blue-difference and red-difference chroma components. Y′ (with prime) is distinguished from Y, which is luminance, meaning that light intensity is nonlinearly encoded based on gamma corrected RGB primaries.

[0033] Chroma subsampling is the practice of encoding images by implementing less resolution for chroma information than for luma information, taking advantage of the human visual system's lower acuity for color differences than for luminance.

1.1 The 4:4:4 color format

[0034] Each of the three Y'CbCr components have the same sample rate, thus there is no chroma subsampling. This scheme is sometimes used in high-end film scanners and cinematic post production.

1.2 The 4:2:2 color format

[0035] The two chroma components are sampled at half the sample rate of luma, e.g. the horizontal chroma resolution is halved. This reduces the bandwidth of an uncompressed video signal by one-third with little to no visual difference.

1.3 The 4:2:0 color format

[0036] In 4:2:0, the horizontal sampling is doubled compared to 4:1:1, but as the Cb and Cr channels are only sampled on each alternate line in this scheme, the vertical

resolution is halved. The data rate is thus the same. Cb and Cr are each subsampled at a factor of 2 both horizontally and vertically. There are three variants of 4:2:0 schemes, having different horizontal and vertical siting.

[0037] ○ In MPEG-2, Cb and Cr are cosited horizontally. Cb and Cr are sited between pixels in the vertical direction (sited interstitially).

[0038] ○ In JPEG/JFIF, H.261, and MPEG-1, Cb and Cr are sited interstitially, halfway between alternate luma samples.

[0039] ○ In 4:2:0 DV, Cb and Cr are co-sited in the horizontal direction. In the vertical direction, they are co-sited on alternating lines.

2 Examples of the coding flow of a typical video codec

[0040] FIG. 1 shows an example of encoder block diagram of VVC, which contains three in-loop filtering blocks: deblocking filter (DF), sample adaptive offset (SAO) and ALF. Unlike DF, which uses predefined filters, SAO and ALF utilize the original samples of the current picture to reduce the mean square errors between the original samples and the reconstructed samples by adding an offset and by applying a finite impulse response (FIR) filter, respectively, with coded side information signaling the offsets and filter coefficients. ALF is located at the last processing stage of each picture and can be regarded as a tool trying to catch and fix artifacts created by the previous stages.

3 Examples of a geometry transformation-based adaptive loop filter in JEM

[0041] In the JEM, an geometry transformation-based adaptive loop filter (GALF) with block-based filter adaption is applied. For the luma component, one among 25 filters is selected for each 2×2 block, based on the direction and activity of local gradients.

3.1 Examples of filter shape

[0042] In the JEM, up to three diamond filter shapes (as shown in FIGS. 2A, 2B and 2C for the 5×5 diamond, 7×7 diamond and 9×9 diamond, respectively) can be selected for the luma component. An index is signalled at the picture level to indicate the filter shape used for the luma component. For chroma components in a picture, the 5×5 diamond shape is always used.

3.1.1 Block classification

Each 2×2 block is categorized into one out of 25 classes. The classification index C is derived based on its directionality D and a quantized value of activity $\hat{A}$, as follows:

$$C = 5D + \hat{A}. \quad (1)$$

To calculate D and $\hat{A}$, gradients of the horizontal, vertical and two diagonal direction are first calculated using 1-D Laplacian:

$$g_v = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} V_{k,l},\ V_{k,l} = |2R(k,l) - R(k,l-1) - R(k,l+1)|, \quad (2)$$

$$g_h = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} H_{k,l},\ H_{k,l} = |2R(k,l) - R(k-1,l) - R(k+1,l)|, \quad (3)$$

$$g_{d1} = \sum_{k=i-2}^{i+3} \sum_{l=j-3}^{j+3} D1_{k,l},\ D1_{k,l} = |2R(k,l) - R(k-1,l-1) - R(k+1,l+1)| \quad (4)$$

$$g_{d2} = \sum_{k=i-2}^{i+3} \sum_{j=j-2}^{j+3} D2_{k,l},\ D2_{k,l} = |2R(k,l) - R(k-1,l+1) - R(k+1,l-1)| \quad (5)$$

Indices i and j refer to the coordinates of the upper left sample in the 2×2 block and $R(i,j)$ indicates a reconstructed sample at coordinate (i,j).

Then D maximum and minimum values of the gradients of horizontal and vertical directions are set as:

$$g_{h,v}^{max} = max(g_h, g_v),\ \ g_{h,v}^{min} = min(g_h, g_v), \quad (6)$$

and the maximum and minimum values of the gradient of two diagonal directions are set as:

$$g_{d0,d1}^{max} = max(g_{d0}, g_{d1}),\ \ g_{d0,d1}^{min} = min(g_{d0}, g_{d1}), \quad (7)$$

To derive the value of the directionality D, these values are compared against each other and with two thresholds t_1 and t_2:

Step 1. If both $g_{h,v}^{max} \le t_1 \cdot g_{h,v}^{min}$ and $g_{d0,d1}^{max} \le t_1 \cdot g_{d0,d1}^{min}$ are true, D is set to 0.

Step 2. If $g_{h,v}^{max}/g_{h,v}^{min} > g_{d0,d1}^{max}/g_{d0,d1}^{min}$, continue from Step 3; otherwise continue from Step 4.

Step 3. If $g_{h,v}^{max} > t_2 \cdot g_{h,v}^{min}$, D is set to 2; otherwise D is set to 1.

Step 4. If $g_{d0,d1}^{max} > t_2 \cdot g_{d0,d1}^{min}$, D is set to 4; otherwise D is set to 3.

The activity value A is calculated as:

$$A = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} (V_{k,l} + H_{k,l}). \quad (8)$$

A is further quantized to the range of 0 to 4, inclusively, and the quantized value is denoted as $\hat{A}$.

For both chroma components in a picture, no classification method is applied, i.e. a single set of ALF coefficients is applied for each chroma component.

3.1.2 Geometric transformations of filter coefficients

[0043] Before filtering each 2×2 block, geometric transformations such as rotation or diagonal and vertical flipping are applied to the filter coefficients $f(k,l)$ depending on gradient

values calculated for that block. This is equivalent to applying these transformations to the samples in the filter support region. The idea is to make different blocks to which ALF is applied more similar by aligning their directionality.

[0044] Three geometric transformations, including diagonal, vertical flip and rotation are introduced:

$$\begin{aligned} &\text{Diagonal: } f_D(k,l) = f(l,k), \\ &\text{Vertical flip: } f_V(k,l) = f(k,K-l-1), \\ &\text{Rotation: } f_R(k,l) = f(K-l-1,k). \end{aligned} \tag{9}$$

[0045] Herein, K is the size of the filter and $0 \le k,l \le K-1$ are coefficients coordinates, such that location $(0,0)$ is at the upper left corner and location $(K-1,K-1)$ is at the lower right corner. The transformations are applied to the filter coefficients *f* (*k*, *l*) depending on gradient values calculated for that block. The relationship between the transformation and the four gradients of the four directions are summarized in Table 1.

Table 1: Mapping of the gradient calculated for one block and the transformations

Gradient values	Transformation
$g_{d2} < g_{d1}$ and $g_h < g_v$	No transformation
$g_{d2} < g_{d1}$ and $g_v < g_h$	Diagonal
$g_{d1} < g_{d2}$ and $g_h < g_v$	Vertical flip
$g_{d1} < g_{d2}$ and $g_v < g_h$	Rotation

3.1.3 Signaling of filter parameters

[0046] In the JEM, GALF filter parameters are signaled for the first CTU, i.e., after the slice header and before the SAO parameters of the first CTU. Up to 25 sets of luma filter coefficients could be signaled. To reduce bits overhead, filter coefficients of different classification can be merged. Also, the GALF coefficients of reference pictures are stored and allowed to be reused as GALF coefficients of a current picture. The current picture may choose to use GALF coefficients stored for the reference pictures, and bypass the GALF coefficients signaling. In this case, only an index to one of the reference pictures is signaled, and the stored GALF coefficients of the indicated reference picture are inherited for the current picture.

[0047] To support GALF temporal prediction, a candidate list of GALF filter sets is maintained. At the beginning of decoding a new sequence, the candidate list is empty. After decoding one picture, the corresponding set of filters may be added to the candidate list. Once the size of the candidate list reaches the maximum allowed value (i.e., 6 in current JEM), a new set of filters overwrites the oldest set in decoding order, and that is, first-in-first-out (FIFO) rule is

applied to update the candidate list. To avoid duplications, a set could only be added to the list when the corresponding picture doesn't use GALF temporal prediction. To support temporal scalability, there are multiple candidate lists of filter sets, and each candidate list is associated with a temporal layer. More specifically, each array assigned by temporal layer index (TempIdx) may compose filter sets of previously decoded pictures with equal to lower TempIdx. For example, the k-th array is assigned to be associated with TempIdx equal to k, and it only contains filter sets from pictures with TempIdx smaller than or equal to k. After coding a certain picture, the filter sets associated with the picture will be used to update those arrays associated with equal or higher TempIdx.

[0048] Temporal prediction of GALF coefficients is used for inter coded frames to minimize signaling overhead. For intra frames, temporal prediction is not available, and a set of 16 fixed filters is assigned to each class. To indicate the usage of the fixed filter, a flag for each class is signaled and if required, the index of the chosen fixed filter. Even when the fixed filter is selected for a given class, the coefficients of the adaptive filter f(k,l) can still be sent for this class in which case the coefficients of the filter which will be applied to the reconstructed image are sum of both sets of coefficients.

[0049] The filtering process of luma component can controlled at CU level. A flag is signaled to indicate whether GALF is applied to the luma component of a CU. For chroma component, whether GALF is applied or not is indicated at picture level only.

3.1.4 Filtering process

[0050] At decoder side, when GALF is enabled for a block, each sample $R(i,j)$ within the block is filtered, resulting in sample value $R'(i,j)$ as shown below, where L denotes filter length, $f_{m,n}$ represents filter coefficient, and $f(k,l)$ denotes the decoded filter coefficients.

$$R'(i,j) = \sum_{k=-L/2}^{L/2} \sum_{l=-L/2}^{L/2} f(k,l) \times R(i+k, j+l) \quad (10)$$

3.1.5 Determination process for encoder side filter parameters

[0051] Overall encoder decision process for GALF is illustrated in FIG. 3. For luma samples of each CU, the encoder makes a decision on whether or not the GALF is applied and the appropriate signalling flag is included in the slice header. For chroma samples, the decision to apply the filter is done based on the picture-level rather than CU-level. Furthermore, chroma GALF for a picture is checked only when luma GALF is enabled for the picture.

4 Examples of a geometry transformation-based adaptive loop filter in VVC

[0052] The current design of GALF in VVC has the following major changes compared to that in JEM:

1) The adaptive filter shape is removed. Only 7x7 filter shape is allowed for luma component and 5x5 filter shape is allowed for chroma component.
2) Temporal prediction of ALF parameters and prediction from fixed filters are both removed.
3) For each CTU, one bit flag is signaled whether ALF is enabled or disabled.
4) Calculation of class index is performed in 4x4 level instead of 2x2. In addition, as proposed in JVET-L0147, sub-sampled Laplacian calculation method for ALF classification is utilized, as shown in FIGS. 4A−4D. More specifically, there is no need to calculate the horizontal/vertical/45 diagonal /135 degree gradients for each sample within one block. Instead, 1:2 subsampling is utilized.

[0053] In VTM4.0, the filtering process of the Adaptive Loop Filter, is performed as follows:

[0054] $$O(x,y) = \sum_{(i,j)} w(i,j).I(x+i,y+j) \quad (11)$$

[0055] where samples $I(x+i,y+j)$ are input samples, $O(x,y)$ is the filtered output sample (i.e. filter result), and $w(i,j)$ denotes the filter coefficients. In practice, in VTM4.0 it is implemented using integer arithmetic for fixed point precision computations:

[0056] $$O(x,y) = \left(\sum_{i=-\frac{L}{2}}^{\frac{L}{2}}\sum_{j=-\frac{L}{2}}^{\frac{L}{2}} w(i,j).I(x+i,y+j) + 64\right) \gg 7 \quad (12)$$

[0057] where *L* denotes the filter length, and where $w(i,j)$ are the filter coefficients in fixed point precision.

5 Non-linear adaptive loop filtering (ALF) in JVET-N0242

5.1 Filtering reformulation

[0058] Equation (11) can be reformulated, without coding efficiency impact, in the following expression:

[0059] $$O(x,y) = I(x,y) + \sum_{(i,j)\neq(0,0)} w(i,j).\big(I(x+i,y+j) - I(x,y)\big) \quad (13)$$

[0060] Herein, $w(i,j)$ are the same filter coefficients as in equation (11) [excepted $w(0,0)$ which is equal to 1 in equation (13) while it is equal to $1 - \sum_{(i,j)\neq(0,0)} w(i,j)$ in equation (11)].

5.2 Modified filter

[0061] Using this above filter formula of (13), we can easily introduce non linearity to make ALF more efficient by using a simple clipping function to reduce the impact of neighbor sample values ($I(x+i,y+j)$) when they are too different with the current sample value ($I(x,y)$) being filtered.

[0062] In this proposal, the ALF filter is modified as follows:

[0063] $$O'(x,y) = I(x,y) + \sum_{(i,j)\neq(0,0)} w(i,j) \times K(I(x+i,y+j) - I(x,y), k(i,j)) \quad (14)$$

[0064] Herein, $K(d,b) = \min(b, \max(-b,d))$ is the clipping function, and $k(i,j)$ are clipping parameters, which depends on the(i,j) filter coefficient. The encoder performs the optimization to find the best $k(i,j)$. Note that for implementation in integer precision, shifting with rounding $\sum_{(i,j)\neq(0,0)} w(i,j) \times K(I(x+i,y+j) - I(x,y), k(i,j))$ is applied.

[0065] In the JVET-N0242 implementation, the clipping parameters $k(i,j)$ are specified for each ALF filter, one clipping value is signaled per filter coefficient. It means that up to 12 clipping values can be signalled in the bitstream per Luma filter and up to 6 clipping values for the Chroma filter.

[0066] In order to limit the signaling cost and the encoder complexity, we limit the evaluation of the clipping values to a small set of possible values. In the proposal, we only use 4 fixed values which are the same for INTER and INTRA tile groups.

[0067] Because the variance of the local differences is often higher for Luma than for Chroma, we use two different sets for the Luma and Chroma filters. We also include the maximum sample value (here 1024 for 10 bits bit-depth) in each set, so that clipping can be disabled if it is not necessary.

[0068] The sets of clipping values used in the JVET-N0242 tests are provided in the Table 2. The 4 values have been selected by roughly equally splitting, in the logarithmic domain, the full range of the sample values (coded on 10 bits) for Luma, and the range from 4 to 1024 for Chroma.

[0069] More precisely, the Luma table of clipping values have been obtained by the following formula:

[0070] $\text{AlfClip}_L = \left\{ \text{round}\left(\left((M)^{\frac{1}{N}} \right)^{N-n+1} \right) \text{ for } n \in 1..N \right]\right\}$, with M=$2^{10}$ and N=4.

[0071] Similarly, the Chroma tables of clipping values is obtained according to the following formula:

[0072] $\text{AlfClip}_C = \left\{ \text{round}\left(A.\left(\left(\frac{M}{A}\right)^{\frac{1}{N-1}} \right)^{N-n} \right) \text{ for } n \in 1..N \right]\right\}$, with M=$2^{10}$, N=4 and A=4.

Table 2: Authorized clipping values

	INTRA/INTER tile group
LUMA	{ 1024, 181, 32, 6 }
CHROMA	{ 1024, 161, 25, 4 }

[0073] The selected clipping values are coded in the “alf_data” syntax element by using a Golomb encoding scheme corresponding to the index of the clipping value in the above Table 2.

This encoding scheme is the same as the encoding scheme for the filter index.

5.2.1 Syntax, semantics

[0074] The newly introduced syntax changes (shown below in bolded, italicized, and underlined font) due to the NLALF is

7.3.4.3 Adaptive loop filter data syntax

alf_data( ) {	**Descriptor**
alf_chroma_idc	*tu(v)*
alf_luma_clip	***u(1)***
if(alf_choma_idc)	
alf_chroma_clip	***u(1)***
alf_luma_num_filters_signalled_minus1	tb(v)
if(alf_luma_num_filters_signalled_minus1 > 0) {	
for(filtIdx = 0; filtIdx < NumAlfFilters; filtIdx++)	
alf_luma_coeff_delta_idx[filtIdx]	tb(v)
}	
alf_luma_coeff_delta_flag	u(1)
if (!alf_luma_coeff_delta_flag && alf_luma_num_filters_signalled_minus1 > 0)	
alf_luma_coeff_delta_prediction_flag	u(1)
alf_luma_min_eg_order_minus1	ue(v)
for(i = 0; i < 3; i++)	
alf_luma_eg_order_increase_flag[i]	u(1)

if (alf_luma_coeff_delta_flag) {	
for(sigFiltIdx = 0; sigFiltIdx <= alf_luma_num_filters_signalled_minus1; sigFiltIdx++)	
alf_luma_coeff_flag[sigFiltIdx]	u(1)
}	
for(sigFiltIdx = 0; sigFiltIdx <= alf_luma_num_filters_signalled_minus1; sigFiltIdx++) {	
if (alf_luma_coeff_flag[sigFiltIdx]) {	
for (j = 0; j < 12; j++) {	
alf_luma_coeff_delta_abs[sigFiltIdx][j]	uek(v)
if(alf_luma_coeff_delta_abs[sigFiltIdx][j])	
alf_luma_coeff_delta_sign[sigFiltIdx][j]	u(1)
}	
}	
}	
if(alf_luma_clip) {	
alf_luma_clip_min_eg_order_minus1	***ue(v)***
for(i = 0; i < 3; i++)	
alf_luma_clip_eg_order_increase_flag[i]	***u(1)***
for (sigFiltIdx = 0; sigFiltIdx <= alf_luma_num_filters_signalled_minus1; sigFiltIdx++) {	
if (alf_luma_coeff_flag[sigFiltIdx]) {	
for (j = 0; j < 12; j++) {	
if(filterCoefficients[sigFiltIdx][j])	
alf_luma_clip_idx[sigFiltIdx][j]	***uek(v)***
}	
}	
}	
}	
if (alf_chroma_idc > 0) {	
alf_chroma_min_eg_order_minus1	ue(v)
for(i = 0; i < 2; i++)	
alf_chroma_eg_order_increase_flag[i]	u(1)
for(j = 0; j < 6; j++) {	
alf_chroma_coeff_abs[j]	uek(v)
if(alf_chroma_coeff_abs[j] > 0)	
alf_chroma_coeff_sign[j]	u(1)
}	
}	
if (alf_chroma_idc > 0 && alf_chroma_clip) {	

alf_chroma_clip_min_eg_order_minus1	***ue(v)***
for(i = 0; i < 2; i++)	
alf_chroma_clip_eg_order_increase_flag[i]	***u(1)***
for(j = 0; j < 6; j++) {	
if(alf_chroma_coeff_abs[j])	
alf_chroma_clip_idx[j]	***uek(v)***
}	
}	
}	

6 CTU-based ALF in JVET-N0427

[0075] Adaptive parameter set (APS) was adopted in VTM4. Each APS contains one set of signalled ALF filters, up to 32 APSs are supported. In the proposal, slice-level temporal filter is tested. A tile group can re-use the ALF information from an APS to reduce the overhead. The APSs are updated as a first-in-first-out (FIFO) buffer.

[0076] For luma component, when ALF is applied to a luma CTB, the choice among 16 fixed, 5 temporal or 1 signaled filter sets (signalled in slice level) is indicated. Only the filter set index is signalled. For one slice, only one new set of 25 filters can be signaled. If a new set is signalled for a slice, all the luma CTBs in the same slice share that set. Fixed filter sets can be used to predict the new slice-level filter set and can be used as candidate filter sets for a luma CTB as well. The number of filters is 64 in total.

[0077] For chroma component, when ALF is applied to a chroma CTB, if a new filter is signalled for a slice, the CTB used the new filter, otherwise, the most recent temporal chroma filter satisfying the temporal scalability constrain is applied.

[0078] As the slice-level temporal filter, the APSs are updated as a first-in-first-out (FIFO) buffer.

7 Post-reconstruction filters

7.1 Diffusion filter (DF)

[0079] In JVET-L0157, diffusion filter is proposed, wherein the intra/inter prediction signal of the CU may be further modified by diffusion filters.

[0080] Uniform diffusion filter. The Uniform Diffusion Filter is realized by convolving the prediction signal with a fixed mask that is either given as h^I or as h^{IV}, defined below.
Besides the prediction signal itself, one line of reconstructed samples left and above of the block are used as an input for the filtered signal, where the use of these reconstructed samples can

be avoided on inter blocks.

[0081] Let $pred$ be the prediction signal on a given block obtained by intra or motion compensated prediction. In order to handle boundary points for the filters, the prediction signal needs to be extended to a prediction signal $pred_{ext}$. This extended prediction can be formed in two ways:

[0082] Either, as an intermediate step, one line of reconstructed samples left and above the block are added to the prediction signal and then the resulting signal is mirrored in all directions. Or only the prediction signal itself is mirrored in all directions. The latter extension is used for inter blocks. In this case, only the prediction signal itself comprises the input for the extended prediction signal $pred_{ext}$.

[0083] If the filter h^I is to be used, it is proposed to replace the prediction signal $pred$ by

[0084]
$$h^I * pred,$$

[0085] using the aforementioned boundary extension. Here, the filter mask h^I is given as

[0086]
$$h^I = (0.25)^4 \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 6 & 0 & 16 & 0 & 6 & 0 & 0 \\ 0 & 4 & 0 & 24 & 0 & 24 & 0 & 4 & 0 \\ 1 & 0 & 16 & 0 & 36 & 0 & 16 & 0 & 1 \\ 0 & 4 & 0 & 24 & 0 & 24 & 0 & 4 & 0 \\ 0 & 0 & 6 & 0 & 16 & 0 & 6 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

[0087] If the filter h^{IV} is to be used, it is proposed to replace the prediction signal $pred$ by

[0088]
$$h^{IV} * pred$$

[0089] Here, the filter h^{IV} is given as

$$h^{IV} = h^I * h^I * h^I * h^I.$$

[0090] Directional diffusion filter. Instead of using signal adaptive diffusion filters, directional filters, a horizontal filter h^{hor} and a vertical filter h^{ver}, are used which still have a fixed mask. More precisely, the uniform diffusion filtering corresponding to the mask h^I of the previous section is simply restricted to be either applied only along the vertical or along the horizontal direction. The vertical filter is realized by applying the fixed filter mask

[0091] $$h_{ver} = (0.5)^4 \begin{pmatrix} 1 \\ 0 \\ 4 \\ 0 \\ 6 \\ 0 \\ 4 \\ 0 \\ 1 \end{pmatrix}$$

[0092] to the prediction signal and the horizontal filter is realized by using the transposed mask $h_{hor} = h_{ver}^t$.

7.2 Bilateral filter (BF)

[0093] Bilateral filter is proposed in JVET-L0406, and it is always applied to luma blocks with non-zero transform coefficients and slice quantization parameter larger than 17. Therefore, there is no need to signal the usage of the bilateral filter. Bilateral filter, if applied, is performed on decoded samples right after the inverse transform. In addition, the filter parameters, i.e., weights are explicitly derived from the coded information.

[0094] The filtering process is defined as:

[0095] $$P'_{0,0} = P_{0,0} + \sum_{k=1}^{K} W_k\big(abs(P_{k,0} - P_{0,0})\big) \times \big(P_{k,0} - P_{0,0}\big). \quad (1)$$

[0096] Herein, $P_{0,0}$ is the intensity of the current sample and $P'_{0,0}$ is the modified intensity of the current sample, $P_{k,0}$ and W_k are the intensity and weighting parameter for the *k*-th neighboring sample, respectively. An example of one current sample and its four neighboring samples (i.e., K=4) is depicted in FIG. 5.

[0097] More specifically, the weight $W_k(x)$ associated with the *k*-th neighboring sample is defined as follows:

[0098] $$W_k(x) = Distance_k \times Range_k(x). \quad (2)$$

[0099] Herein,

[00100] $$Distance_k = e^{\left(-\frac{10000}{2\sigma_d^2}\right)} \Big/ 1 + 4 * e^{\left(-\frac{10000}{2\sigma_d^2}\right)} \text{ and } Range_k(x) = e^{\left(-\frac{x^2}{8*(QP-17)*(QP-17)}\right)}.$$

[00101] Herein, σ_d is dependent on the coded mode and coding block sizes. The described filtering process is applied to intra-coded blocks, and inter-coded blocks when TU is further split, to enable parallel processing.

[00102] To better capture statistical properties of video signal, and improve performance of the filter, weights function resulted from Equation (2) are being adjusted by the σ_d parameter, tabulated in Table 4 as being dependent on coding mode and parameters of block partitioning (minimal size).

Table 4: Value of σ_d for different block sizes and coding modes

Min(block width, block height)	Intra mode	Inter mode
4	82	62
8	72	52
Other	52	32

[00103] To further improve the coding performance, for inter-coded blocks when TU is not split, the intensity difference between current sample and one of its neighboring samples is replaced by a representative intensity difference between two windows covering current sample and the neighboring sample. Therefore, the equation of filtering process is revised to:

[00104] $$P'_{0,0} = P_{0,0} + \sum_{k=1}^{N} W_k \left(\frac{1}{M}\sum_{m=-M/2}^{M/2} abs(P_{k,m} - P_{0,m})\right) \times (P_{k,0} - P_{0,0}) \quad (4)$$

[00105] Herein, $P_{k,m}$ and $P_{0,m}$ represent the *m*-th sample value within the windows centered at $P_{k,0}$ and $P_{0,0}$, respectively. In this proposal, the window size is set to 3×3. An example of two windows covering $P_{2,0}$ and $P_{0,0}$ are depicted in FIG. 6.

7.3 Hadamard transform domain filter (HF)

[00106] In JVET-K0068, in-loop filter in 1D Hadamard transform domain which is applied on CU level after reconstruction and has multiplication free implementation. Proposed filter is applied for all CU blocks that meet the predefined condition and filter parameters are derived from the coded information.

[00107] Proposed filtering is always applied to luma reconstructed blocks with non-zero transform coefficients, excluding 4x4 blocks and if slice quantization parameter is larger than 17. The filter parameters are explicitly derived from the coded information. Proposed filter, if applied, is performed on decoded samples right after inverse transform.

[00108] For each pixel from reconstructed block pixel processing comprises the following steps:

[00109] ○ Scan 4 neighboring pixels around processing pixel including current one according to scan pattern

[00110] ○ 4 point Hadamard transform of read pixels

[00111] ○ Spectrum filtering based on the following formula:

[00112] $$F(i,\sigma) = \frac{R(i)^2}{R(i)^2+\sigma^2} * R(i)$$

[00113] Herein, *(i)* is index of spectrum component in Hadamard spectrum, *R(i)* is spectrum component of reconstructed pixels corresponding to index, σ is filtering parameter deriving from codec quantization parameter QP using following equation:

[00114] $$\boldsymbol{\sigma = 2^{(1+0.126*(QP-27))}}.$$

[00115] The example of scan pattern is shown in FIG. 7, wherein A is the current pixel and {B,C,D} are surrounding pixels.

[00116] For pixels laying on CU boundary, the scan pattern is adjusted ensuring all required pixels are within current CU.

8 Virtual Pipelining Data Units (VPDU)

[00117] Virtual pipeline data units (VPDUs) are defined as non-overlapping MxM-luma(L)/NxN-chroma(C) units in a picture. In hardware decoders, successive VPDUs are processed by multiple pipeline stages at the same time; different stages process different VPDUs simultaneously. The VPDU size is roughly proportional to the buffer size in most pipeline stages, so it is said to be very important to keep the VPDU size small. In HEVC hardware decoders, the VPDU size is set to the maximum transform block (TB) size. Enlarging the maximum TB size from 32x32-L/16x16-C (as in HEVC) to 64x64-L/32x32-C (as in the current VVC) can bring coding gains, which results in 4X of VPDU size (64x64-L/32x32-C) expectedly in comparison with HEVC. However, in addition to quadtree (QT) coding unit (CU) partitioning, ternary tree (TT) and binary tree (BT) are adopted in VVC for achieving additional coding gains, and TT and BT splits can be applied to 128x128-L/64x64-C coding tree blocks (CTUs) recursively, which is said to lead to 16X of VPDU size (128x128-L/64x64-C) in comparison with HEVC.

[00118] In current design of VVC, the VPDU size is defined as 64x64-L/32x32-C.

9 Drawbacks of existing implementations

[00119] The non-linear ALF (NLALF) design in JVET-N0242 has the following problems:

[00120] (1) The classification process in GALF relies on the gradient and Laplacian activities which utilize the reconstructed samples before applying ALF. However, the classification results may be inaccurate. For example, for one sample, differences between it and its neighbors may be quite similar while for another sample, difference between it and one neighbor may be too different and for all others, difference may be too small. For these two cases, they may be classified to one class index which may be unreasonable.

[00121] (2) The clipping parameters are associated with the filter coefficient. However, one filter may be utilized for multiple classes due to filter merging process. In addition, for two blocks with the same class index (0…24 in current GALF design), the filter coefficients and clipping parameters are the same. However, the two blocks may have different characteristics, for example, the selected geometric transformation may be different. Using same clipping parameters may be suboptimal.

[00122] (3) The index of clipping parameters is signaled per non-zero filter coefficient which requires to construct the filter coefficients in the parsing stage. Such a design is undesirable for hardware implementation.

[00123] (4) In the temporal prediction process, one block may inherit the filter coefficients from previously coded frames. How to handle the clipping parameters need to be studied.

[00124] (5) The clipping function $K(d,b) = \min(b, \max(-b,d))$ with b as the upper bounder and -b as the lower bound, d as the input. The restriction of equal magnitudes for the upper and lower bounder may be suboptimal.

10 Exemplary methods for temporal prediction of parameters in non-linear ALF

[00125] Embodiments of the presently disclosed technology overcome the drawbacks of existing implementations, thereby providing video coding with higher coding efficiencies. Temporal prediction of parameters in non-linear adaptive loop filtering, based on the disclosed technology, may enhance both existing and future video coding standards, is elucidated in the following examples described for various implementations. The examples of the disclosed technology provided below explain general concepts, and are not meant to be interpreted as limiting. In an example, unless explicitly indicated to the contrary, the various features described in these examples may be combined.

[00126] In these examples, one filter may be associated with multiple filter coefficients. One set of filters represents multiple filters. Denote the i-th filter by F^i and its associated filter coefficients by F^ik, such as the variable k denotes the k-th filter coefficient associated with F^i, e.g., it may be corresponding to Ck in FIG. 2.

1. It is proposed to determine the NLALF parameters (e.g., on/off control flag, clipping parameters) according to coded information.
    a. The NLALF parameters (e.g., on/off control flag, clipping parameters) may be dependent on coded mode information.
        i. In one example, which NLALF parameters to be selected may be determined by the coded mode, such as intra or non-intra mode.
        ii. In one example, which NLALF parameters to be selected may be determined by the coded mode, such as intra or inter mode.
        iii. In one example, which NLALF parameters to be selected may be determined by the coded mode, such as IBC or non-IBC mode.
    b. The NLALF parameters (e.g., on/off control flag, clipping parameters) may be dependent on transform information.

i. In one example, they may be dependent on whether transform skip is applied or not.

c. The NLALF parameters (e.g., on/off control flag, clipping parameters) may be dependent on residual information.
   i. In one example, they may be dependent on whether the block contains non-zero coefficients.

d. The NLALF parameters (e.g., on/off control flag, clipping parameters) may be dependent on tile group types/picture types.

e. The NLALF parameters (e.g., on/off control flag, clipping parameters) may be dependent on temporal layer information/reference picture information associated with one tile/tile group/slice etc. al.
   i. In one example, they may be dependent on whether all reference pictures are associated with smaller POC values compared to current picture.
   ii. In one example, they may be dependent on whether all reference pictures are associated with smaller or equal POC values compared to current picture.

f. It is proposed to determine the NLALF parameters (e.g., on/off control flag, clipping parameters) according to reference picture/motion information associated with one block.

2. It is proposed to determine the NLALF parameters (e.g., on/obiff control flag, clipping parameters) according to the geometric transformation.
   a. In one example, for two MxN blocks, even they are associated with the same filter (e.g., due to the same class index), the associated NLALF parameters (e.g., clipping parameters) may be different.
   b. In one example, for one filter coefficient, indications of more than one clipping parameter may be signaled.
      i. In one example, how many clipping parameters/or indices of clipping parameters or other representations of clipping parameters may be dependent on the number of allowed geometric transformations.
      ii. In one example, predictive coding of clipping parameters/indices of clipping parameters associated with one filter parameter may be applied.

1) In one example, a clipping parameter of one filter for one sample or block may be predicted by another clipping parameter of another filter used for a spatial/temporal adjacent or non-adjacent neighbouring sample or block.

3. It is proposed that magnitudes of the upper bound and the lower bound in the clipping function may be unequal.
    a. In one example, indications of both upper bound and lower bound for one clipping function may be signaled.
        i. Alternatively, furthermore, predictive coding between upper bound and lower bound may be applied.
4. It is proposed to directly code the indications of clipping parameters (such as indices) with fixed length.
    a. In one example, each of them may be coded with N-bits (e.g., N is set to 2).
        i. In one example, N may be fixed;
        ii. In one example, N may be signaled;
        iii. In one example, N may depend on coding information, such as QP, picture dimensions and so on.
    b. Alternatively, they may be coded with truncated unary method with a maximum value N.
        i. In one example, N may be fixed;
        ii. In one example, N may be signaled;
        iii. In one example, N may depend on coding information, such as QP, picture dimensions and so on.
    c. Alternatively, they may be coded with Exponential-Golomb method but with fixed order for one filter/for one filter set.
    d. Alternatively, they may be coded with run-length coding.
        i. In one example, for each filter, the index of a clipping parameter may be firstly coded as ‘run’, and the number of consecutive same clipping parameter as ‘length’.
        ii. In one example, for each k-th filter coefficient in all filters, the index of a clipping parameter associated with F^i may be firstly coded as ‘run’, and the number of same clipping parameter in other filters as ‘length’.

e. In one example, predictive coding of indications of clipping parameters (such as indices) may be applied.
    i. In one example, predictive coding may be applied for clipping parameters within one filter.
    ii. In one example, predictive coding may be applied for clipping parameters among different filters.
        1) In one example, predictive coding may be applied for clipping parameters among different filters for one color component.
        2) In one example, predictive coding may be applied for clipping parameters among different filters for multiple color component.
        3) In one example, predictive coding may be applied for clipping parameters of filters used for different samples or blocks.
    iii. In one example, predictive coding may be applied for clipping parameters signaled in different APSs.

5. It is proposed to decouple the parsing of clipping parameters and construction of filter coefficients.
    a. In one example, the parsing of clipping parameters (e.g., index of clipping parameters) is independent from the values of filter coefficients.
    b. In one example, when the filter coefficient is equal to 0, indication of the associated clipping parameter may be still signaled.
6. When one block inherits filter coefficients from the i-th filter, the associated clipping parameters with the i-th filter may be not inherited.
    a. In one example, when temporal prediction is enabled for one block, instead of directly inheriting the associated clipping parameters, whether to use non-local ALF or not (applying clipping or not) may be signaled.
        i. In one example, if it is determined to apply clipping, the associated clipping parameters may be also inherited.
    b. In one example, suppose the filter coefficients are inherited/predicted from the i-th filter, the clipping parameters are inherited/predicted from the j-th filter, i may be unequal to j.
    c. In one example, suppose the filter coefficients are inherited/predicted from the i-th filter, the clipping parameters are inherited/predicted from the j-th filter, the i-th and j-th filter may be associated with different filter set.

i. In one example, the i-th filter may be associated with a first picture/tile group/tile/slice and the j-th filter may be associated with a second picture/tile group/tile/slice.
ii. In one example, i is unequal to j. Alternatively, i is equal to j.

d. In one example, indications of clipping parameters associated with which filter may be signaled, such as the filter index.
e. In one example, indications of clipping parameters associated with which filter set may be signaled, such as the APS index.
i. Alternatively, furthermore, the filter index may be further signalled.

7. In the classification process, instead of directly using the sample difference, the clipped sample difference may be utilized.
a. In one example, in the gradient calculation process, clipped sample difference or clipped gradients may be used.
b. In one example, in the activity calculation process, clipped sample difference or clipped gradients may be used.
c. In one example, the following may be used to calculate the vertical gradient:

$$V_{k,l} = \left|clip1\big(R(k,l) - R(k,l-1)\big) + clip2(R(k,l) - R(k,l+1))\right|$$

wherein clip1 and clip2 are two clipping functions.
d. In one example, the following may be used to calculate the horizontal gradient:

$$H_{k,l} = \left|clip1\big(R(k,l) - R(k-1,l)\big) + clip2(R(k,l) - R(k+1,l))\right|$$

wherein clip1 and clip2 are two clipping functions.

8. Whether to apply the clipping operations may depend on the locations of the samples (such as $I(x+i, y+j)$ in Section 5.2) to be used in the filtering process.
a. In one example, if the sample in a filter support is not located a CU/PU/TU/picture/tile/tile group boundary, clipping may be disabled.
b. In one example, if the sample in a filter support is located a CU/PU/TU/picture/tile/tile group/ CTU/Virtual Pipelining Data Unit (VPDU) boundary, clipping may be applied.
c. Alternatively, whether to apply the clipping operations may depend on the distance of the samples (such as $I(x+i, y+j)$ in Section 5.2) to be used in the filtering process from the CU/PU/TU/picture/tile/tile group/CTU/VPDU boundary.
i. In one example, the distance may be pre-defined (e.g., N-pel).
ii. In one example, the distance may be signaled.

9. The filter shape (a.k.a. filter support) used in adaptive loop filter process may depend on the color representation.
    a. In one example, the filter support for all components (such as Y, Cb, Cr) should be the same when the color format is 4:4:4.
        i. For example, the filter support is 7*7 diamond as shown in FIG. 2B.
        ii. For example, the filter support is 5*5 diamond as shown in FIG. 2A.
    b. In one example, the support area for all components when the color format is RGB.
        i. For example, the filter support is 7*7 diamond as shown in FIG. 2B.
        ii. For example, the filter support is 5*5 diamond as shown in FIG. 2A.

[00127] The examples described above may be incorporated in the context of the method described below, e.g., methods 800, 810 and 820, which may be implemented at a video decoder or a video encoder.

[00128] FIG. 8A shows a flowchart of an exemplary method for video processing. The method 800 includes, at step 802, determining, based on a characteristic of a current video block, one or more parameters of a non-linear filtering operation.

[00129] The method 800 includes, at step 804, performing, based on the one or more parameters, a conversion between the current video block and a bitstream representation of the current video block.

[00130] In some embodiments, the characteristic of the current video block is a coding mode of the current video block. In an example, the coding mode of the current video block is an intra mode, a non-intra mode, an intra block copy (IBC) mode or a non-IBC mode.

[00131] In some embodiments, the characteristic is transform information. In an example, the transform information comprises an indication of transform skip being applied to the current video block.

[00132] In some embodiments, the characteristic is residual information. In an example, the residual information comprises zero-valued coefficients in the current video block.

[00133] In some embodiments, the characteristic is a tile group type or a picture type of a tile group or a picture comprising the current video block.

[00134] In some embodiments, the characteristic is temporal layer information or reference information associated with a tile, a tile group, a picture or a slice comprising the current video block.

[00135] In some embodiments, the characteristic is a reference picture or motion information

associated with the current video block.

[00136] In some embodiments, the characteristic is a geometric transformation.

[00137] In some embodiments, the one or more parameters comprises an on/off control flag or one or more parameters of a clipping function.

[00138] In some embodiments, a magnitude of an upper bound of the clipping function is different from a magnitude of a lower bound of the clipping function. In an example, predictive coding is applied between the upper bound and the lower bound of the clipping function.

[00139] FIG. 8B shows a flowchart of an exemplary method for video processing. The method 810 includes, at step 812, configuring, for a current video block, one or more parameters of a clipping operation that is part of a non-linear filtering operation.

[00140] The method 810 includes, at step 814, performing, based on the one or more parameters, a conversion between the current video block and a bitstream representation of the current video block.

[00141] In some embodiments, the one or more parameters is coded with a fixed length of N bits. In other embodiments, the one or more parameters is coded with a truncated unary method with a maximum value of N. In an example, N is fixed. In another example, N is signaled. In yet another example, N is based on coded information of the current video block that comprises a quantization parameter or a dimension of the picture comprising the current video block.

[00142] In some embodiments, the one or more parameters is coded with an Exponential-Golomb method with a fixed order for one filter or filter set.

[00143] In some embodiments, the one or more parameters is coded with run-length coding.

[00144] In some embodiments, the one or more parameters is signaled independently from values of at least one filter coefficient.

[00145] In some embodiments, the one or more parameters further comprises filter coefficients, wherein the current video block inherits the filter coefficients from an i-th filter, and wherein the one or more parameters of the clipping function are inherited from a j-th filter that is different from the i-th filter.

[00146] FIG. 8C shows a flowchart of an exemplary method for video processing. The method 820 includes, at step 822, configuring a non-linear filtering operation that comprises a clipping operation.

[00147] The method 820 includes, at step 824, performing, based on the non-linear filtering operation, a conversion between a current video block and a bitstream representation of the current video block.

[00148] In some embodiments, the method 820 further includes the step of performing a gradient calculation process that uses a clipped sample difference or a clipped gradient generated using the clipping operation. In other embodiments, the method 820 further includes the step of performing an activity calculation process that uses a clipped sample difference or a clipped gradient generated using the clipping operation. In an example, the clipped gradient comprises a vertical gradient that is computed as $V_{k,l} = \left|clip1\big(R(k,l) - R(k,l-1)\big) + clip2(R(k,l) - R(k,l+1))\right|$. In another example, the clipped gradient comprises a horizontal gradient that is computed as $H_{k,l} = \left|clip1\big(R(k,l) - R(k-1,l)\big) + clip2(R(k,l) - R(k+1,l))\right|$, where clip1 and clip2 are a first and a second clipping function, respectively.

[00149] In some embodiments, performing the conversion comprises filtering one or more samples of the current video block, and an operation of the clipping operation is configured based on a location of the one or more samples.

[00150] In some embodiments, the location of the one or more samples is a boundary of a coding unit (CU), a prediction unit (PU), a transform unit (TU), a picture, a tile, a tile group, a coding tree unit (CTU) or a virtual pipelining data unit (VPDU).

[00151] In some embodiments, and in the context of methods 800, 810 and 820, a shape of a filter used in the non-linear filtering operation is based on a color representation. In an example, the color representation comprises a 4:4:4 color format or an RGB color format. In another example, the filter is a diamond shaped filter.

[00152] In some embodiments, and in the context of methods 800, 810 and 820, the non-linear filtering operation is a non-linear adaptive loop filtering operation.

11 Example implementations of the disclosed technology

[00153] FIG. 9 is a block diagram of a video processing apparatus 900. The apparatus 900 may be used to implement one or more of the methods described herein. The apparatus 900 may be embodied in a smartphone, tablet, computer, Internet of Things (IoT) receiver, and so on. The apparatus 900 may include one or more processors 902, one or more memories 904 and video processing hardware 906. The processor(s) 902 may be configured to implement one or more methods (including, but not limited to, methods 800, 810 and 820) described in the present document. The memory (memories) 904 may be used for storing data and code used for implementing the methods and techniques described herein. The video processing hardware 906 may be used to implement, in hardware circuitry, some techniques described in the present document.

[00154] In some embodiments, the video coding methods may be implemented using an

apparatus that is implemented on a hardware platform as described with respect to FIG. 9.

[00155] FIG. 10 is a block diagram showing an example video processing system 1000 in which various techniques disclosed herein may be implemented. Various implementations may include some or all of the components of the system 1000. The system 1000 may include input 1002 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 1002 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.

[00156] The system 1000 may include a coding component 1004 that may implement the various coding or encoding methods described in the present document. The coding component 1004 may reduce the average bitrate of video from the input 1002 to the output of the coding component 1004 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 1004 may be either stored, or transmitted via a communication connected, as represented by the component 1006. The stored or communicated bitstream (or coded) representation of the video received at the input 1002 may be used by the component 1008 for generating pixel values or displayable video that is sent to a display interface 1010. 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.

[00157] 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.

[00158] FIG. 11 shows a flowchart of an example method for visual media processing. Steps of this flowchart are discussed in connection with example 4 in Section 10 of this document. At step 1102, the process configures, for a current video block, one or more parameters of a clipping operation that is part of a non-linear filtering operation. At step 1104, the process performs, based

on the one or more parameters, a conversion between the current video block and a bitstream representation of the current video block, wherein the one or more parameters is coded in accordance with a rule.

[00159] FIG. 12 shows a flowchart of an example method for visual media processing. Steps of this flowchart are discussed in connection with example 1 in Section 10 of this document. At step 1202, the process determines, based on a characteristic of a current video block, one or more parameters of a non-linear filtering operation. At step 1204, the process performs, based on the one or more parameters, a conversion between the current video block and a bitstream representation of the current video block.

[00160] FIG. 13 shows a flowchart of an example method for visual media processing. Steps of this flowchart are discussed in connection with example 5 in Section 10 of this document. At step 1302, the process configures, for a current video block, one or more parameters of a clipping operation that is part of a non-linear filtering operation. At step 1304, the process performs, based on the one or more parameters, a conversion between the current video block and a bitstream representation of the current video block, wherein the one or more parameters are presented in the bitstream representation independently from values of at least one filter coefficient associated with the non-linear filtering operation.

[00161] FIG. 14 shows a flowchart of an example method for visual media processing. Steps of this flowchart are discussed in connection with example 6 in Section 10 of this document. At step 1402, the process configures, for a current video block, one or more parameters of a clipping operation that is part of a non-linear filtering operation. At step 1404, the process performs, based on the one or more parameters, a conversion between the current video block and a bitstream representation of the current video block, wherein the current video block inherits filter coefficients from an i-th filter, and wherein a first rule associated with inheritance of the one or more parameters of the clipping operation is different from a second rule associated with inheritance of the filter coefficients.

[00162] Various embodiments discussed herein are now presented in clause-based format.

[00163] A1. A method for visual media processing, comprising:

[00164] configuring, for a current video block, one or more parameters of a clipping operation that is part of a non-linear filtering operation; and

[00165] performing, based on the one or more parameters, a conversion between the current video block and a bitstream representation of the current video block, wherein the one or more parameters is coded in accordance with a rule.

[00166] A2. The method of clause A1, wherein the rule specifies coding the one or more parameters with a fixed length of N bits.

[00167] A3. The method of clause A1, wherein the rule specifies coding the one or more parameters with a truncated unary method based on a maximum value of N.

[00168] A4. The method of any one or more of clauses A1-A3, wherein N is fixed.

[00169] A5. The method of any one or more of clauses A1-A3, wherein N is signaled in the bitstream representation.

[00170] A6. The method of any one or more of clauses A1-A3, wherein N is based on coded information of the current video block that comprises a quantization parameter or a dimension of the picture comprising the current video block.

[00171] A7. The method of clause A1, wherein the non-linear filtering operation is based on a filter, wherein the rule specifies coding the one or more parameters with an Exponential-Golomb method with a fixed order for one filter or filter set.

[00172] A8. The method of clause A1, wherein the rule specifies coding the one or more parameters based on a run-length coding method.

[00173] A9. The method of clause A5, wherein the rule further specifies that a run of the run-length coding method corresponds to an index of the one or more parameters and a length of the run-length coding method corresponds to a number of consecutive parameters of the one or more parameters that are same.

[00174] A10. The method of clause A1, wherein the rule specifies coding the one or more parameters based on predictive coding.

[00175] A11. The method of clause A10, wherein the predictive coding is applied for the one or more parameters within one filter.

[00176] A12. The method of clause A10, wherein the predictive coding is applied for the one or more parameters among different filters.

[00177] A13. The method of clause A12, wherein the predictive coding is applied for the one or more parameters among different filters used for one color component.

[00178] A14. The method of clause A12, wherein the predictive coding is applied for the one or more parameters among different filters used for different color components.

[00179] A15. The method of clause A12, wherein the predictive coding is applied for the one or more parameters among different filters used for different samples of the current video block.

[00180] A16. The method of clause A12, wherein the predictive coding is applied for the

one or more parameters among different filters used for different video blocks.

[00181] A17. The method of clause A11, wherein the one or more parameters are included as fields in different adaptive parameters sets (APSs).

[00182] A18. The method of any one or more of clauses A1 to A17, wherein the non-linear filtering operation is an adaptive loop filter (ALF) operation which comprises determining a filter index based on gradient calculations in different directions.

[00183] A19. The method of any one or more of clauses A1 to A17, wherein the one or more parameters includes a clipping index.

[00184] B1. A method for visual media processing, comprising:

[00185] determining, based on a characteristic of a current video block, one or more parameters of a non-linear filtering operation; and

[00186] performing, based on the one or more parameters, a conversion between the current video block and a bitstream representation of the current video block.

[00187] B2. The method of clause B1, wherein the characteristic of the current video block is a coding mode of the current video block.

[00188] B3. The method of clause B2, wherein the coding mode of the current video block is an intra mode, a non-intra mode, an intra block copy (IBC) mode or a non-IBC mode.

[00189] B4. The method of clause B1, wherein the characteristic is transform information.

[00190] B5. The method of clause B4, wherein the transform information comprises an indication of transform skip being applied to the current video block.

[00191] B6. The method of clause B1, wherein the characteristic is residual information.

[00192] B7. The method of clause B6, wherein the residual information comprises zero-valued coefficients in the current video block.

[00193] B8. The method of clause B1, wherein the characteristic is a tile group type or a picture type of a tile group or a picture comprising the current video block.

[00194] B9. The method of clause B1, wherein the characteristic is temporal layer information or reference information associated with a tile, a tile group, a picture or a slice comprising the current video block.

[00195] B10. The method of clause B1, wherein the characteristic is a reference picture or motion information associated with the current video block.

[00196] B11. The method of clause B1, wherein the characteristic is a geometric transformation.

[00197] B12. The method of any one or more of clauses B1 to B11, wherein the one or more parameters comprises an on/off control flag and/or one or more parameters of a clipping function.

[00198] B13. The method of clause B12, wherein a magnitude of an upper bound of the clipping function is different from a magnitude of a lower bound of the clipping function.

[00199] B14. The method of clause B13, wherein predictive coding is applied between the upper bound and the lower bound of the clipping function.

[00200] B15. The method of clause B12, wherein the upper bound of the clipping function and the lower bound of the clipping function are included as fields in the bitstream representation.

[00201] B16. The method of clause B10, wherein the non-linear filtering operation includes use of a first filter and a second filter, and wherein the one or more parameters of the second filter are predicted using the one or more parameters of the first filter.

[00202] B17. The method of clause B10, wherein the first filter and the second filter are applied on different sets of samples of the current video block.

[00203] B18. The method of clause B10, wherein the first filter and the second filter are applied on samples associated with different video blocks.

[00204] B19. The method of any one or more of clauses A1 to B18, wherein a shape of a filter used in the non-linear filtering operation is based on a color representation of a sample associated with the current video block.

[00205] B20. The method of clause B19, wherein the color representation comprises a 4:4:4 color format or an RGB color format.

[00206] B21. The method of clause B19, wherein the filter is a diamond shaped filter.

[00207] B22. The method of clause B19, wherein the diamond shaped filter is of size 5x5 or 7x7.

[00208] B23. The method of any one or more of clauses B1 to B22, wherein the non-linear filtering operation is a non-linear adaptive loop filtering operation.

[00209] B24. The method of any one or more of clauses B1 to B23, wherein the non-linear filtering operation is an adaptive loop filter (ALF) operation which comprises determining a filter index based on gradient calculations in different directions.

[00210] B25. The method of any one or more of clauses B1 to B23, wherein the one or more parameters includes a clipping index.

[00211] B26. The method of any of clauses A1-B25, wherein the conversion includes generating the bitstream representation from the current video block.

[00212] B27. The method of any of clauses A1-B25, wherein the conversion includes generating pixel values of the current video block from the bitstream representation.

[00213] B28. A video encoder apparatus comprising a processor configured to implement a method recited in any one or more of clauses A1-B25.

[00214] B29. A video decoder apparatus comprising a processor configured to implement a method recited in any one or more of clauses A1-B25.

[00215] B30. A computer readable medium having code stored thereon, the code embodying processor-executable instructions for implementing a method recited in any of or more of clauses A1-B25.

[00216] C1. A method for visual media processing, comprising:

[00217] configuring, for a current video block, one or more parameters of a clipping operation that is part of a non-linear filtering operation; and

[00218] performing, based on the one or more parameters, a conversion between the current video block and a bitstream representation of the current video block, wherein the one or more parameters are presented in the bitstream representation independently from values of at least one filter coefficient associated with the non-linear filtering operation.

[00219] C2. The method of clause C1, wherein the one or more parameters are presented in the bitstream representation in case that a value of the at least one filter coefficient is zero.

[00220] C3. The method of any one or more of clauses C1-C2, wherein the one or more parameters are presented in the bitstream representation regardless of the value of the at least one filter coefficient.

[00221] C4. The method of any one or more of clauses C1 to C3, wherein the non-linear filtering operation is an adaptive loop filter (ALF) operation which comprises determining a filter index based on gradient calculations in different directions.

[00222] C5. The method of any one or more of clauses C1 to C3, wherein the one or more parameters includes a clipping index.

[00223] D1. A method for visual media processing, comprising:

[00224] configuring, for a current video block, one or more parameters of a clipping operation that is part of a non-linear filtering operation; and

[00225] performing, based on the one or more parameters, a conversion between the current video block and a bitstream representation of the current video block, wherein the current video block inherits filter coefficients from an i-th filter, and wherein a first rule associated

with inheritance of the one or more parameters of the clipping operation is different from a second rule associated with inheritance of the filter coefficients.

[00226] D2. The method of clause D1, wherein the first rule specifies excluding inheritance of the one or more parameters of the clipping operation from the i-th filter.

[00227] D3. The method of clause D1, further comprising:

[00228] upon identifying that temporal prediction is enabled for the current video block, making a determination of whether to apply or disable the clipping operation.

[00229] D4. The method of clause D1, further comprising:

[00230] upon identifying that temporal prediction is enabled for the current video block, making a determination of whether to apply or exclude inheritance of the one or more parameters of the clipping operation.

[00231] D5. The method of clause D1, wherein the first rule specifies inheriting the one or more parameters of the clipping operation from a j-th filter.

[00232] D6. The method of clause D1, wherein the first rule specifies inheriting the one or more parameters of the clipping operation from the j-th filter, and wherein the j-th filter and the i-th filter are associated with different filter sets.

[00233] D7. The method of clause D6, wherein the j-th filter and the i-th filter are associated with different pictures and/or tile groups and/or tiles and/or slices.

[00234] D8. The method of clause D6, wherein the j-th filter and the i-th filter are same.

[00235] D9. The method of clause D5, wherein the j-th filter and the i-th filter are different.

[00236] D10. The method of clause D1, wherein the first rule specifies including the one or more parameters of the clipping operation as fields in the bitstream representation.

[00237] D11. The method of clause D10, wherein the fields include an adaptive parameter set (APS) index.

[00238] D12. The method of any one or more of clauses D1-D11, wherein the clipping operation includes computing a clipped sample difference or a clipping gradient.

[00239] D13. The method of clause D12, wherein the clipped gradient comprises a vertical gradient that is computed as

[00240] $V\_(k,l)=|clip1(R(k,l)-R(k,l-1))+clip2(R(k,l)-R(k,l+1))|$, wherein clip1 and clip2 are a first and a second clipping function, respectively and R(i, j) denotes a sample of the current video block.

[00241] D14. The method of clause D12, wherein the clipped gradient comprises a

horizontal gradient that is computed as

[00242] H_(k,l)=|clip1(R(k,l)-R(k-1,l))+clip2(R(k,l)-R(k+1,l))|, wherein clip1 and clip2 are a first and a second clipping function, respectively and R(i, j) denotes a sample of the current video block.

[00243] D15. The method of any one or more of clauses D1 to D14, further comprising:

[00244] making a determination, based on a location of a sample used in the filtering operation, of whether to selectively enable or disable the clipping operation.

[00245] D16. The method of clause D15, wherein the clipping operation is disabled if the sample is not located at a boundary of one or more of the following: a coding unit, a partition unit, a transform unit, a picture, a tile, or a tile group.

[00246] D17. The method of clause D15, wherein the clipping operation is enabled if the sample is located at a boundary of one or more of the following: a coding unit, a partition unit, a transform unit, a picture, a tile, a tile group, a coding tree unit, or a virtual pipelining data unit.

[00247] D18. The method of clause D15, wherein the location is with respect to a distance between the sample and a boundary of one or more of the following: a coding unit, a partition unit, a transform unit, a picture, a tile, a tile group, a coding tree unit, or a virtual pipelining data unit.

[00248] D19. The method of clause D18, wherein the distance is pre-defined.

[00249] D20. The method of clause D18, wherein the distance is signaled in the bitstream representation.

[00250] D21. The method of any one or more of clauses D1 to D20, wherein a shape of a filter used in the non-linear filtering operation is based on a color representation of a sample associated with the current video block.

[00251] D22. The method of clause D21, wherein the color representation comprises a 4:4:4 color format or an RGB color format.

[00252] D23. The method of clause D21, wherein the filter is a diamond shaped filter.

[00253] D24. The method of clause D23, wherein the diamond shaped filter is of size 5x5 or 7x7.

[00254] D25. The method of any one or more of clauses D1 to D24, wherein the non-linear filtering operation is an adaptive loop filtering (ALF) operation which comprises determining a filter index based on gradient calculations in different directions.

[00255] D26. The method of any one or more of clauses D1 to D24, wherein the one or more parameters includes a clipping index.

[00256] D27. The method of any of clauses C1-D26, wherein the conversion includes generating the bitstream representation from the current video block.

[00257] D28. The method of any of clauses C1-D26, wherein the conversion includes generating pixel values of the current video block from the bitstream representation.

[00258] D29. A video encoder apparatus comprising a processor configured to implement a method recited in any one or more of clauses C1-D26.

[00259] D30. A video decoder apparatus comprising a processor configured to implement a method recited in any one or more of clauses C1-D26.

[00260] D31. A computer readable medium having code stored thereon, the code embodying processor-executable instructions for implementing a method recited in any of or more of clauses C1-D26.

[00261] In the present document, the term "video processing" or "visual media processing" or "processing of visual media" may refer to video encoding, video decoding, video compression or video decompression. For example, video compression algorithms may be applied during conversion from pixel representation of a video to a corresponding bitstream representation or vice versa. The bitstream representation of a current video block may, for example, correspond to bits that are either co-located or spread in different places within the bitstream, as is defined by the syntax. For example, a macroblock may be encoded in terms of transformed and coded error residual values and also using bits in headers and other fields in the bitstream. Furthermore, during conversion, a decoder may parse a bitstream with the knowledge that some fields may be present, or absent, based on the determination, as is described in the above solutions. Similarly, an encoder may determine that certain syntax fields are or are not to be included and generate the coded representation accordingly by including or excluding the syntax fields from the coded representation.

[00262] From the foregoing, it will be appreciated that specific embodiments of the presently disclosed technology have been described herein for purposes of illustration, but that various modifications may be made without deviating from the scope of the invention. Accordingly, the presently disclosed technology is not limited except as by the appended claims.

[00263] Implementations of the subject matter and the functional operations described in this patent document can be implemented in various systems, digital electronic circuitry, or in computer software, firmware, or hardware, including the structures disclosed in this specification and their structural equivalents, or in combinations of one or more of them. Implementations of the subject matter described in this specification can be implemented as one or more computer

program products, i.e., one or more modules of computer program instructions encoded on a tangible and non-transitory 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 of them. The term "data processing unit" or "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.

[00264] 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.

[00265] The processes and logic flows described in this specification 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).

[00266] 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 nonvolatile memory, media and memory devices, including by way of example semiconductor memory devices, e.g., EPROM, EEPROM, and flash memory devices. The processor and the memory can be supplemented by, or incorporated in, special purpose logic circuitry.

[00267] It is intended that the specification, together with the drawings, be considered exemplary only, where exemplary means an example. As used herein, the singular forms "a", "an" and "the" are intended to include the plural forms as well, unless the context clearly indicates otherwise. Additionally, the use of "or" is intended to include "and/or", unless the context clearly indicates otherwise.

[00268] While this patent document contains many specifics, these should not be construed as limitations on the scope of any invention or of what may be claimed, but rather as descriptions of features that may be specific to particular embodiments of particular inventions. 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.

[00269] 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.

[00270] 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

What is claimed is:

1. A method for visual media processing, comprising:

configuring, for a current video block, one or more parameters of a clipping operation that is part of a non-linear filtering operation; and

performing, based on the one or more parameters, a conversion between the current video block and a bitstream representation of the current video block,

wherein the one or more parameters are presented in the bitstream representation independently from values of at least one filter coefficient associated with the non-linear filtering operation.

2. The method of claim 1, wherein the one or more parameters are presented in the bitstream representation in case that a value of the at least one filter coefficient is zero.

3. The method of any one or more of claims 1-2, wherein the one or more parameters are presented in the bitstream representation regardless of the value of the at least one filter coefficient.

4. The method of any one or more of claims 1 to 3, wherein the non-linear filtering operation is an adaptive loop filter (ALF) operation which comprises determining a filter index based on gradient calculations in different directions.

5. The method of any one or more of claims 1 to 3, wherein the one or more parameters includes a clipping index.

6. A method for visual media processing, comprising:

configuring, for a current video block, one or more parameters of a clipping operation that is part of a non-linear filtering operation; and

performing, based on the one or more parameters, a conversion between the current video block and a bitstream representation of the current video block, wherein the current video block inherits filter coefficients from an i-th filter, and wherein a first rule associated with inheritance of the one or more parameters of the clipping operation is different from a second rule associated with inheritance of the filter coefficients.

7. The method of claim 6, wherein the first rule specifies excluding inheritance of the one or more parameters of the clipping operation from the i-th filter.

8. The method of claim 6, further comprising:

upon identifying that temporal prediction is enabled for the current video block, making a determination of whether to apply or disable the clipping operation.

9. The method of claim 6, further comprising:

upon identifying that temporal prediction is enabled for the current video block, making a determination of whether to apply or exclude inheritance of the one or more parameters of the clipping operation.

10. The method of claim 6, wherein the first rule specifies inheriting the one or more parameters of the clipping operation from a j-th filter.

11. The method of claim 6, wherein the first rule specifies inheriting the one or more parameters of the clipping operation from the j-th filter, and wherein the j-th filter and the i-th filter are associated with different filter sets.

12. The method of claim 11, wherein the j-th filter and the i-th filter are associated with different pictures and/or tile groups and/or tiles and/or slices.

13. The method of claim 11, wherein the j-th filter and the i-th filter are same.

14. The method of claim 10, wherein the j-th filter and the i-th filter are different.

15. The method of claim 6, wherein the first rule specifies including the one or more parameters of the clipping operation as fields in the bitstream representation.

16. The method of claim 15, wherein the fields include an adaptive parameter set (APS) index.

17. The method of any one or more of claims 6-16, wherein the clipping operation includes computing a clipped sample difference or a clipping gradient.

18. The method of claim 17, wherein the clipped gradient comprises a vertical gradient that is computed as

$$V\_(k,l)=|clip1(R(k,l)-R(k,l-1))+clip2(R(k,l)-R(k,l+1))|,$$

wherein clip1 and clip2 are a first and a second clipping function, respectively and R(i, j) denotes a sample of the current video block.

19. The method of claim 17, wherein the clipped gradient comprises a horizontal gradient that is computed as

H_(k,l)=|clip1(R(k,l)-R(k-1,l))+clip2(R(k,l)-R(k+1,l))|,

wherein clip1 and clip2 are a first and a second clipping function, respectively and R(i, j) denotes a sample of the current video block.

20. The method of any one or more of claims 6 to 19, further comprising:

making a determination, based on a location of a sample used in the filtering operation, of whether to selectively enable or disable the clipping operation.

21. The method of claim 20, wherein the clipping operation is disabled if the sample is not located at a boundary of one or more of the following: a coding unit, a partition unit, a transform unit, a picture, a tile, or a tile group.

22. The method of claim 20, wherein the clipping operation is enabled if the sample is located at a boundary of one or more of the following: a coding unit, a partition unit, a transform unit, a picture, a tile, a tile group, a coding tree unit, or a virtual pipelining data unit.

23. The method of claim 20, wherein the location is with respect to a distance between the sample and a boundary of one or more of the following: a coding unit, a partition unit, a transform unit, a picture, a tile, a tile group, a coding tree unit, or a virtual pipelining data unit.

24. The method of claim 23, wherein the distance is pre-defined.

25. The method of claim 23, wherein the distance is signaled in the bitstream representation.

26. The method of any one or more of claims 6 to 25, wherein a shape of a filter used in the non-linear filtering operation is based on a color representation of a sample associated with the current video block.

27. The method of claim 26, wherein the color representation comprises a 4:4:4 color format or an RGB color format.

28. The method of claim 26, wherein the filter is a diamond shaped filter.

29. The method of claim 28, wherein the diamond shaped filter is of size 5x5 or 7x7.

30. The method of any one or more of claims 6 to 29, wherein the non-linear filtering operation is an adaptive loop filtering (ALF) operation which comprises determining a filter index based on gradient calculations in different directions.

31. The method of any one or more of claims 6 to 29, wherein the one or more parameters includes a clipping index.

32. The method of any of claims 1-31, wherein the conversion includes generating the bitstream representation from the current video block.

33. The method of any of claims 1-31, wherein the conversion includes generating pixel values of the current video block from the bitstream representation.

34. A video encoder apparatus comprising a processor configured to implement a method recited in any one or more of claims 1-31.

35. A video decoder apparatus comprising a processor configured to implement a method recited in any one or more of claims 1-31.

36. A computer readable medium having code stored thereon, the code embodying processor-executable instructions for implementing a method recited in any of or more of claims 1-31.

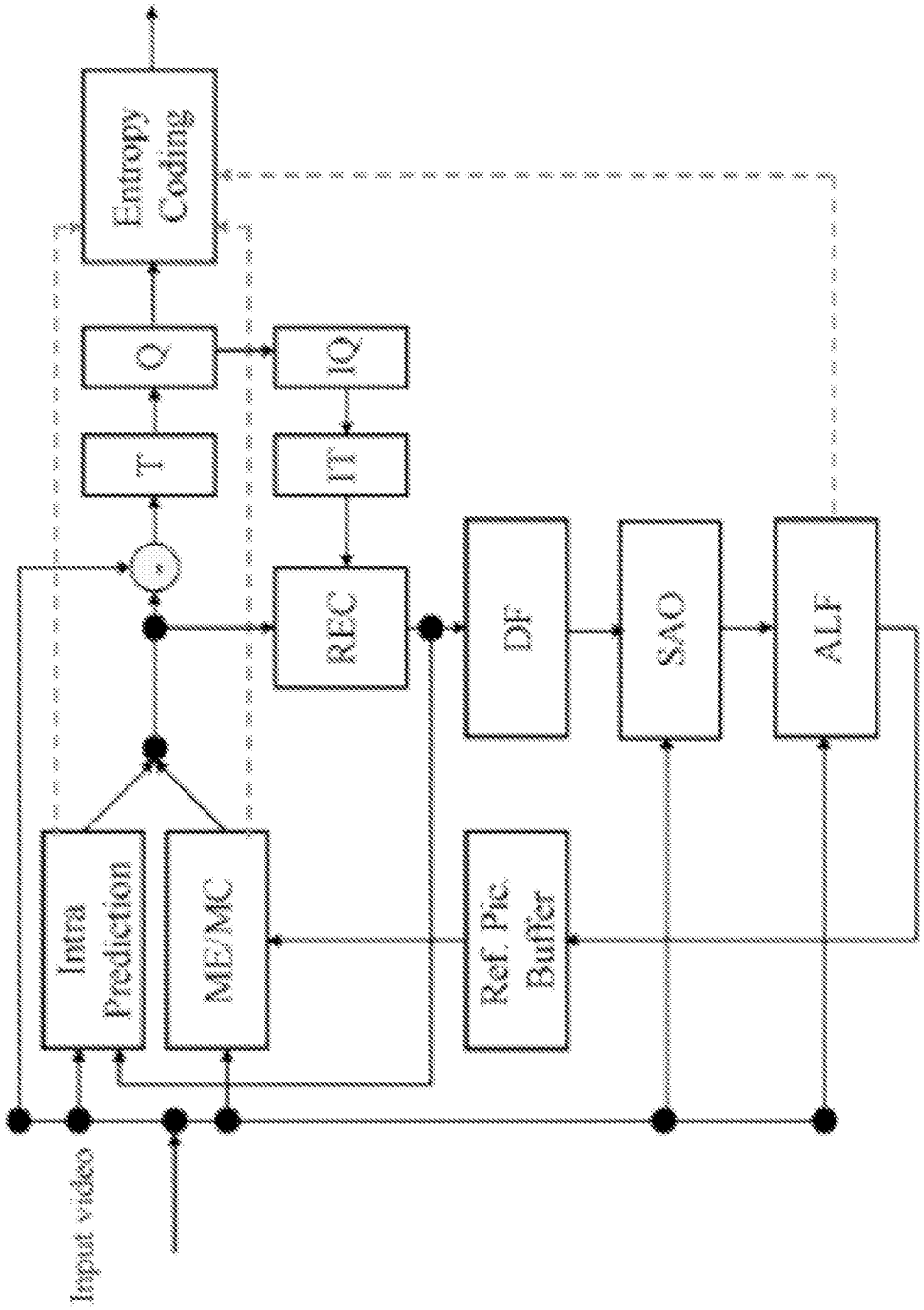
Entropy Coding
Q
IQ
T
IT
REC
DF
SAO
ALF
Intra Prediction
ME/MC
Ref. Pic. Buffer
Input video

FIG. 1

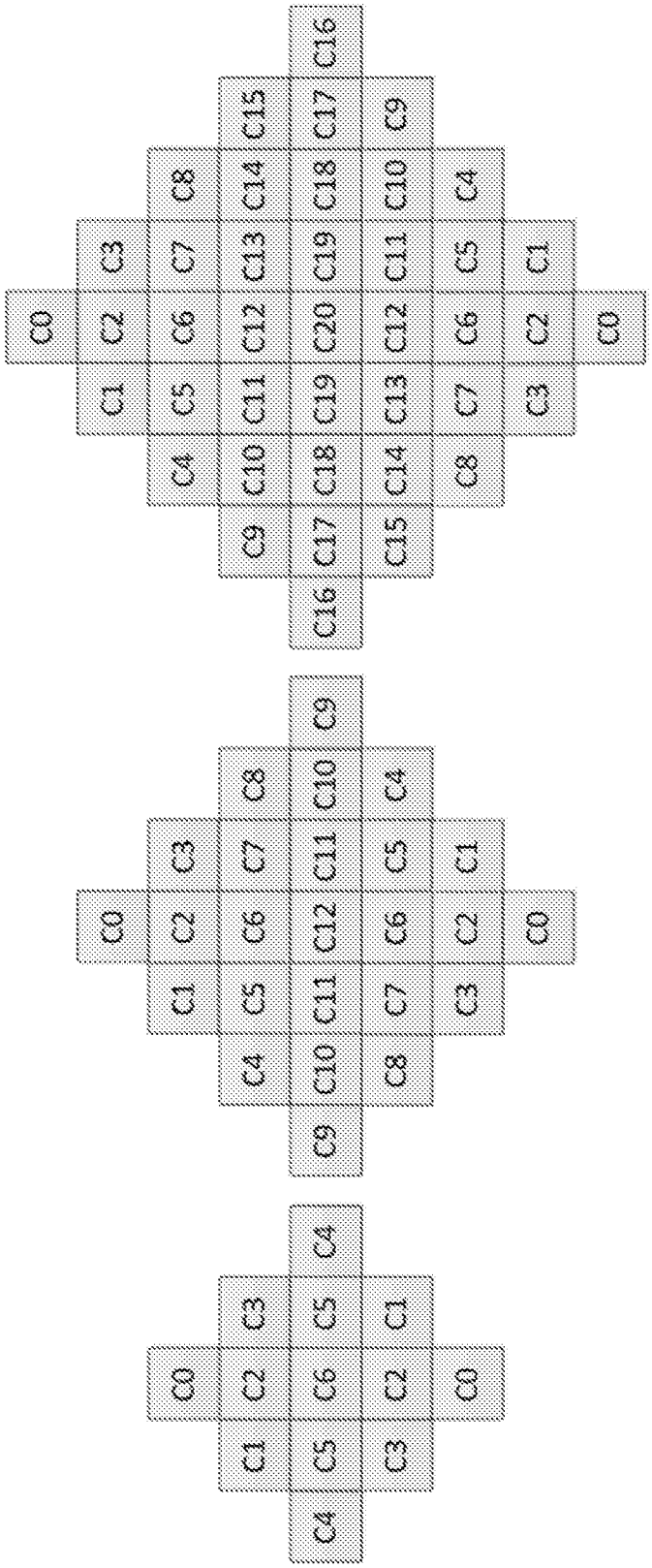

FIG. 2A

FIG. 2B

FIG. 2C

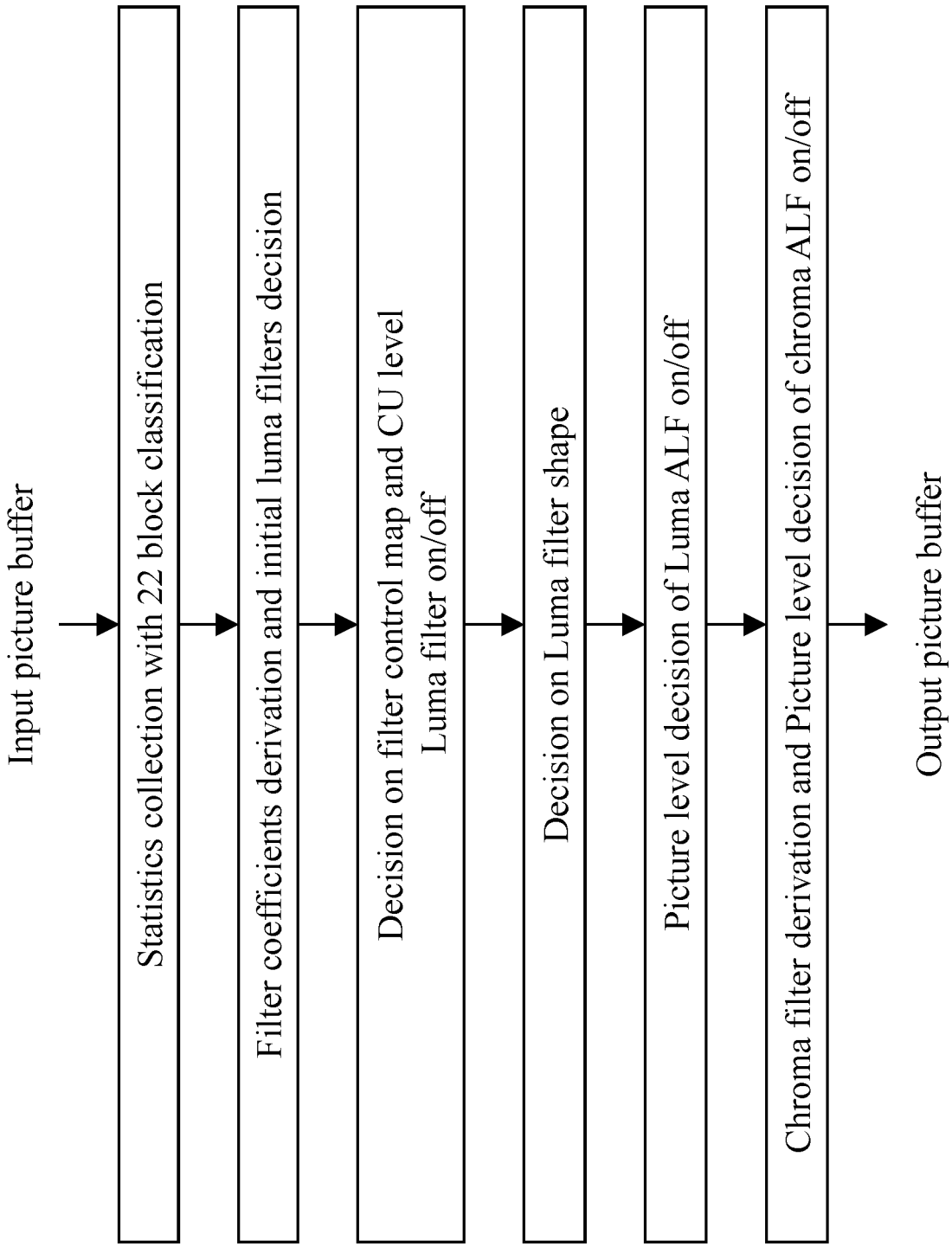
Input picture buffer
Statistics collection with 22 block classification
Filter coefficients derivation and initial luma filters decision
Decision on filter control map and CU level Luma filter on/off
Decision on Luma filter shape
Picture level decision of Luma ALF on/off
Chroma filter derivation and Picture level decision of chroma ALF on/off
Output picture buffer

FIG. 3

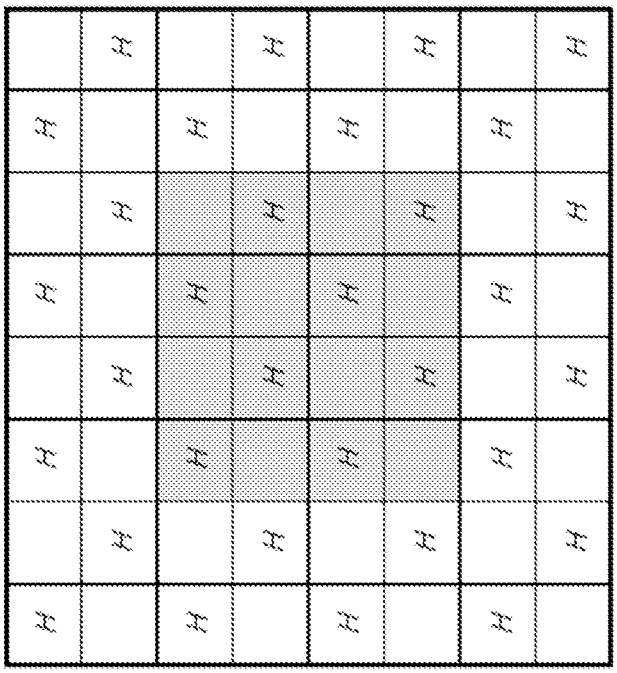

FIG. 4B

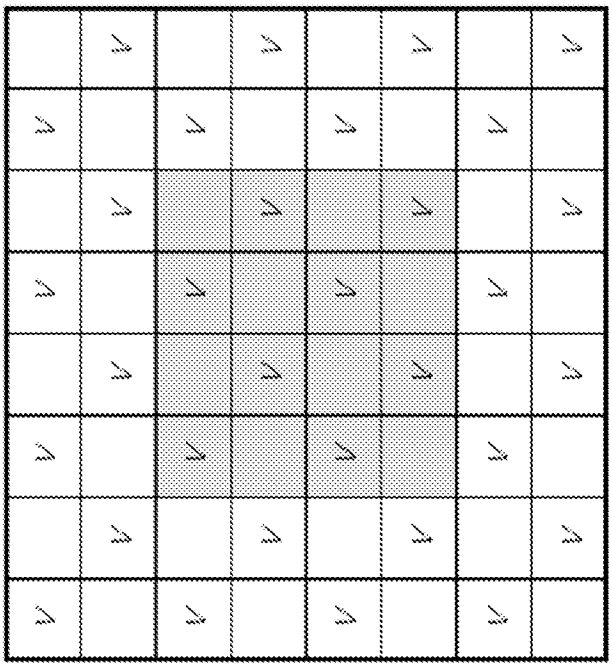

FIG. 4A

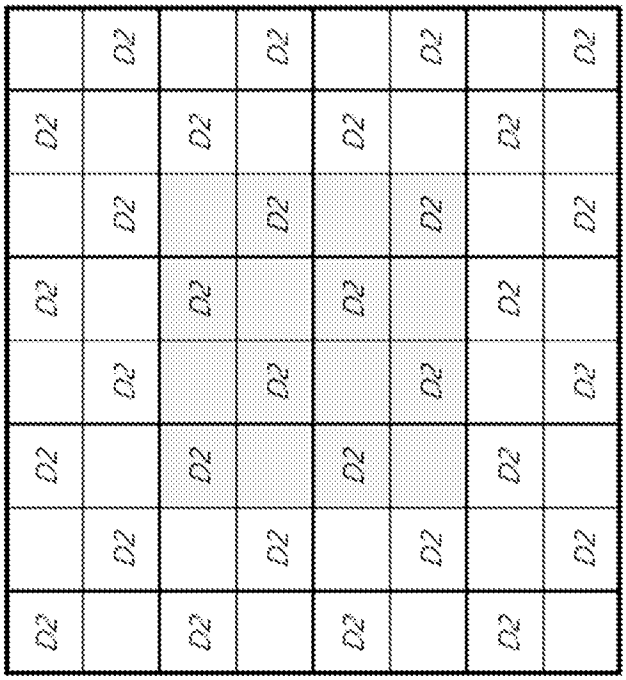

FIG. 4D

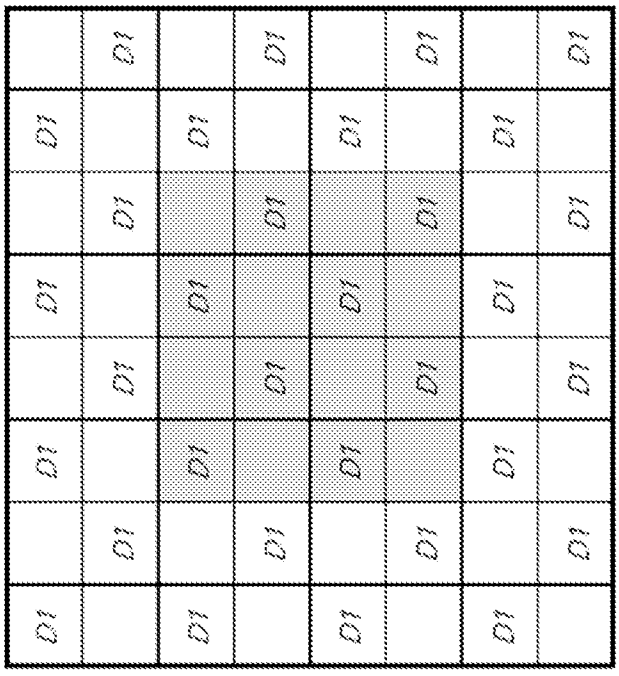

FIG. 4C

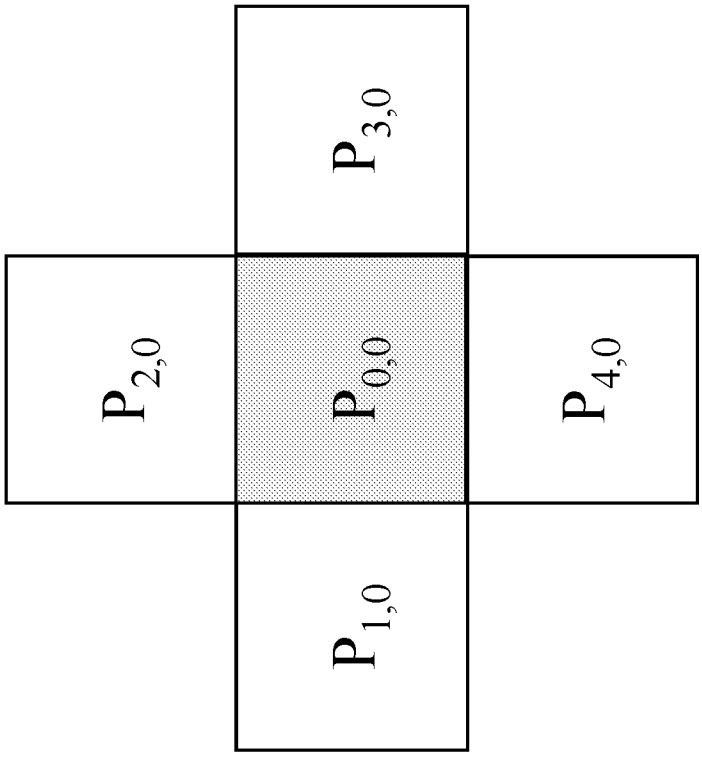

FIG. 5

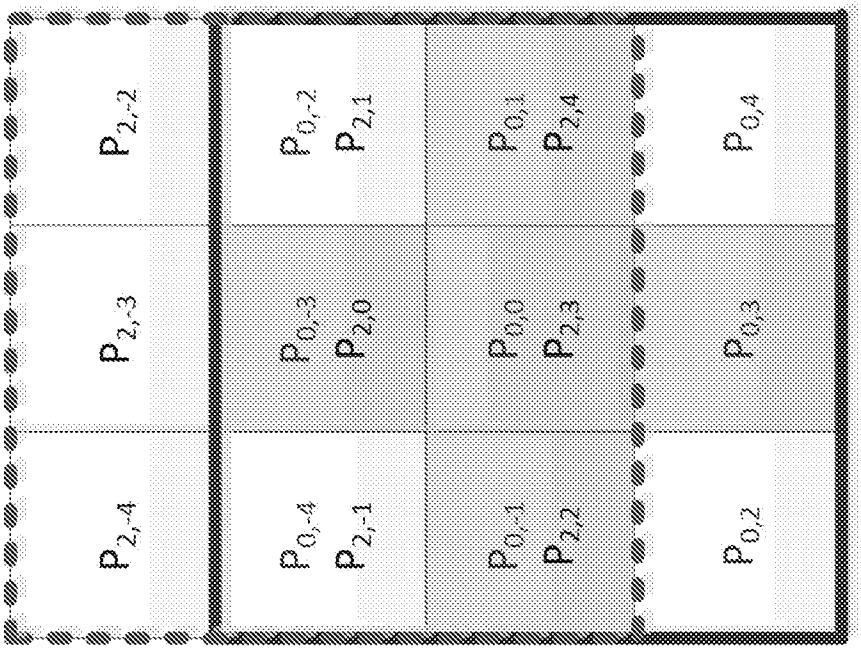
Window associated with the sample $P_{0,0}$
Window associated with the sample $P_{2,0}$
$P_{2,-4}$
$P_{2,-3}$
$P_{2,-2}$
$P_{0,-4}$ $P_{2,-1}$
$P_{0,-3}$ $P_{2,0}$
$P_{0,-2}$ $P_{2,1}$
$P_{0,-1}$ $P_{2,2}$
$P_{0,0}$ $P_{2,3}$
$P_{0,1}$ $P_{2,4}$
$P_{0,2}$
$P_{0,3}$
$P_{0,4}$

FIG. 6

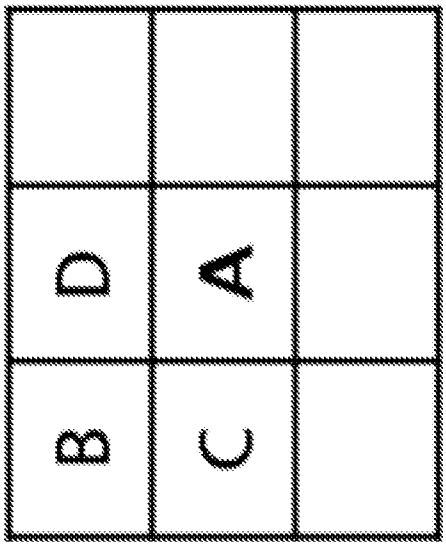

FIG. 7

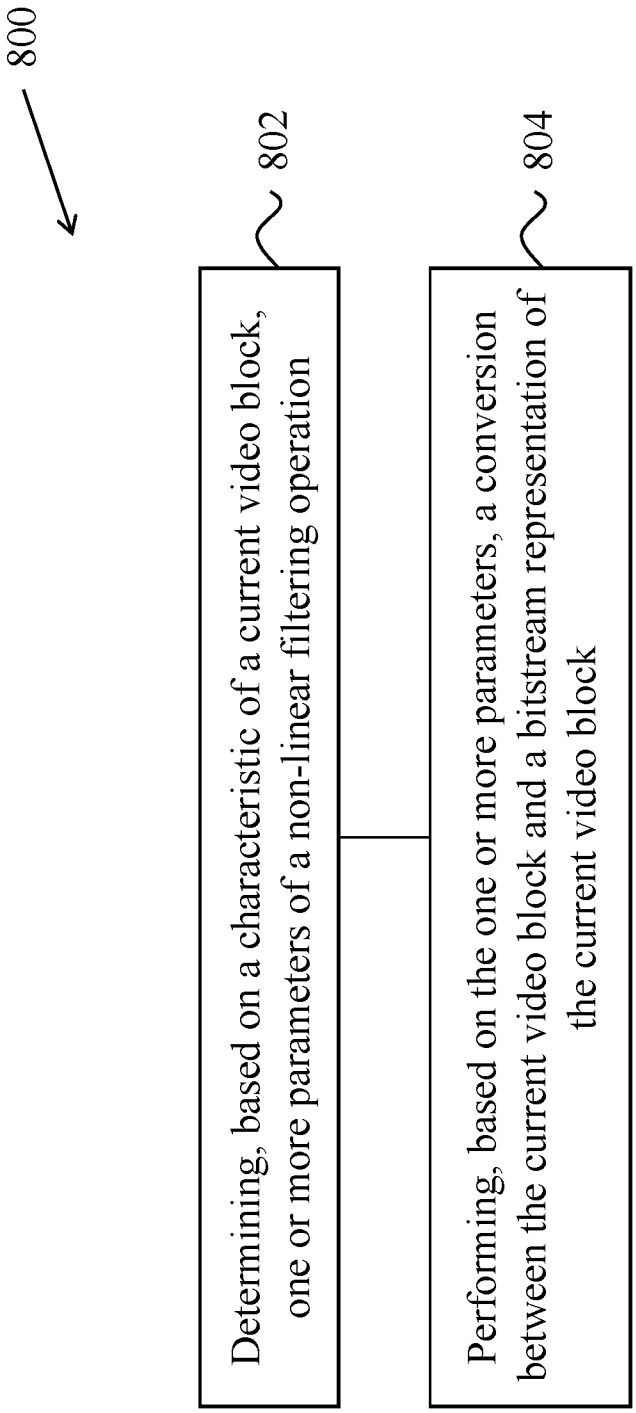
800
802
Determining, based on a characteristic of a current video block, one or more parameters of a non-linear filtering operation
804
Performing, based on the one or more parameters, a conversion between the current video block and a bitstream representation of the current video block

FIG. 8A

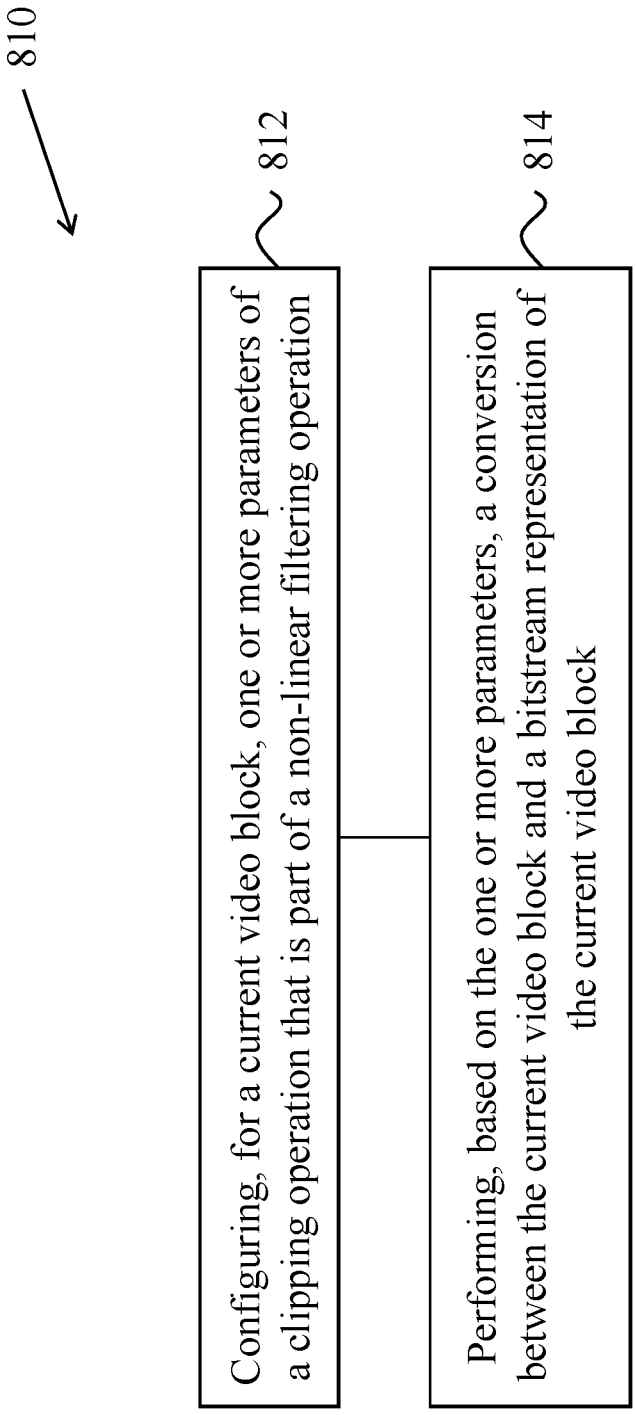
810
812
Configuring, for a current video block, one or more parameters of a clipping operation that is part of a non-linear filtering operation
814
Performing, based on the one or more parameters, a conversion between the current video block and a bitstream representation of the current video block

FIG. 8B

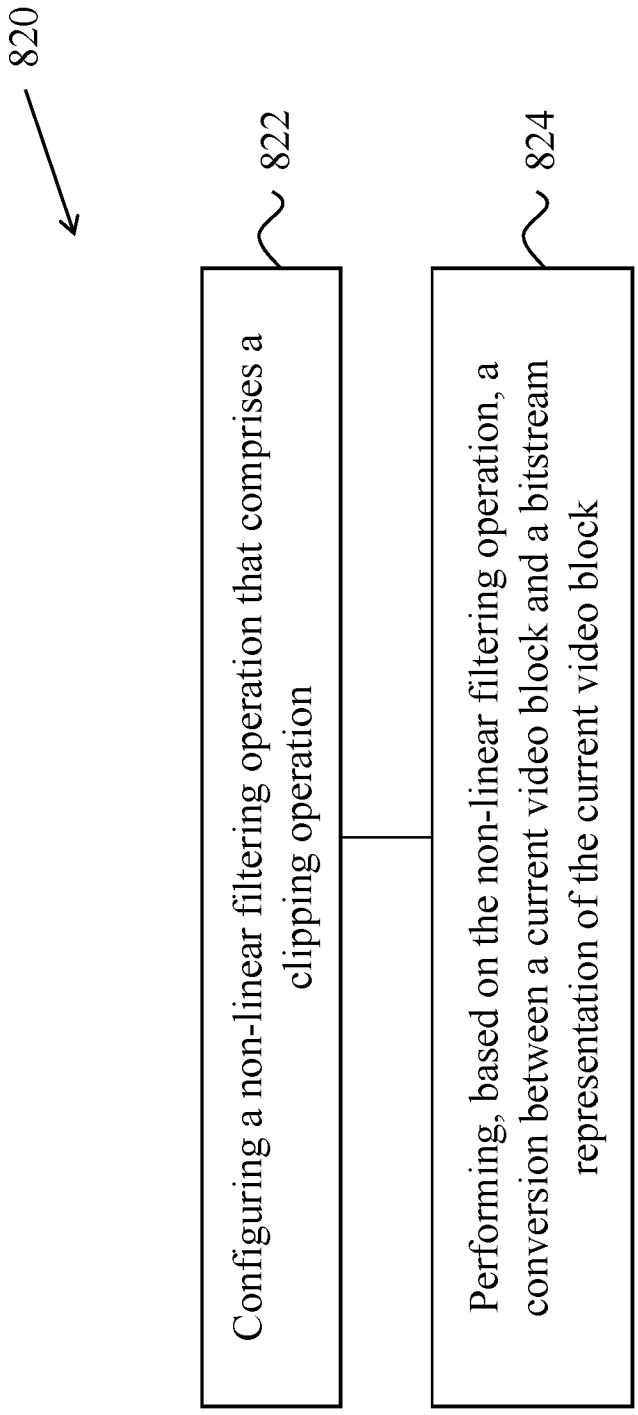
820
822
Configuring a non-linear filtering operation that comprises a clipping operation
824
Performing, based on the non-linear filtering operation, a conversion between a current video block and a bitstream representation of the current video block

FIG. 8C

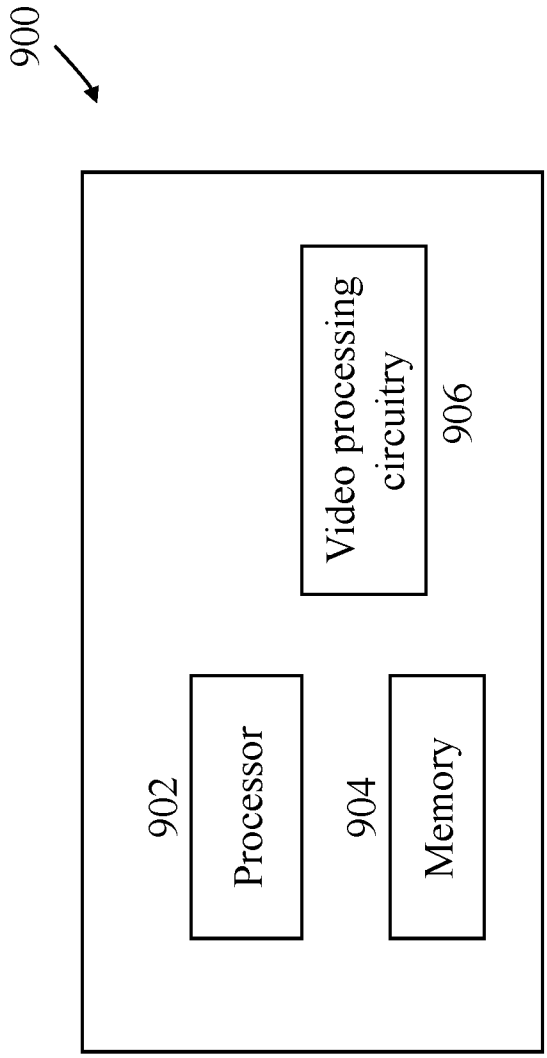
900
Video processing circuitry
906
902
Processor
904
Memory

FIG. 9

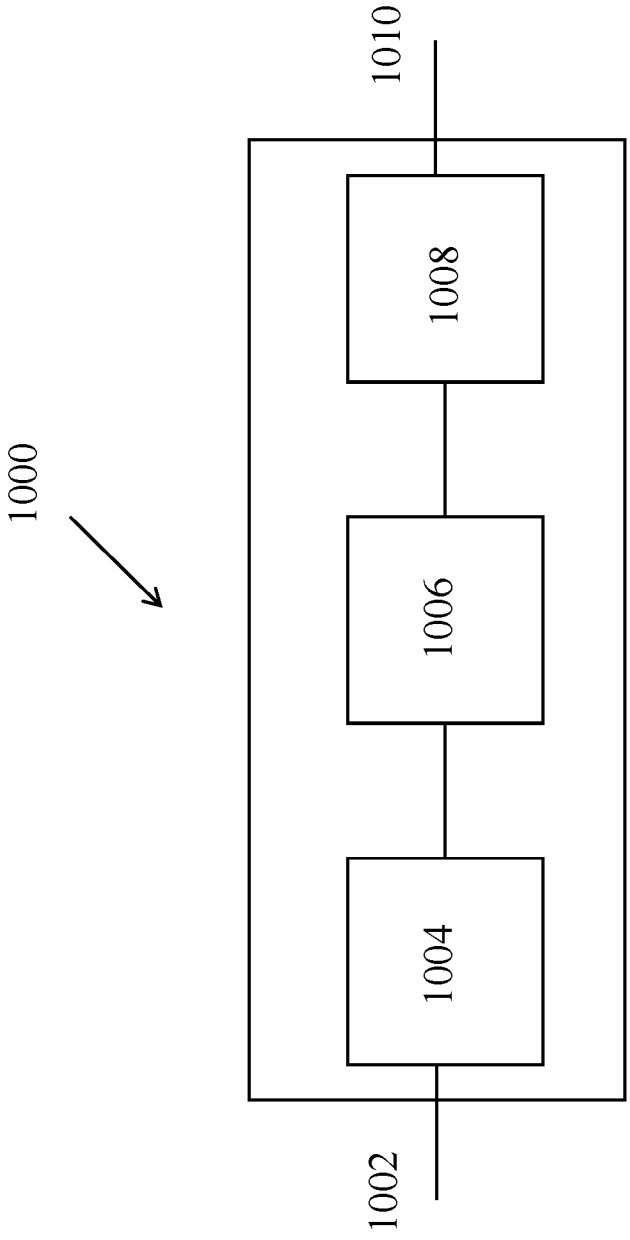

FIG. 10

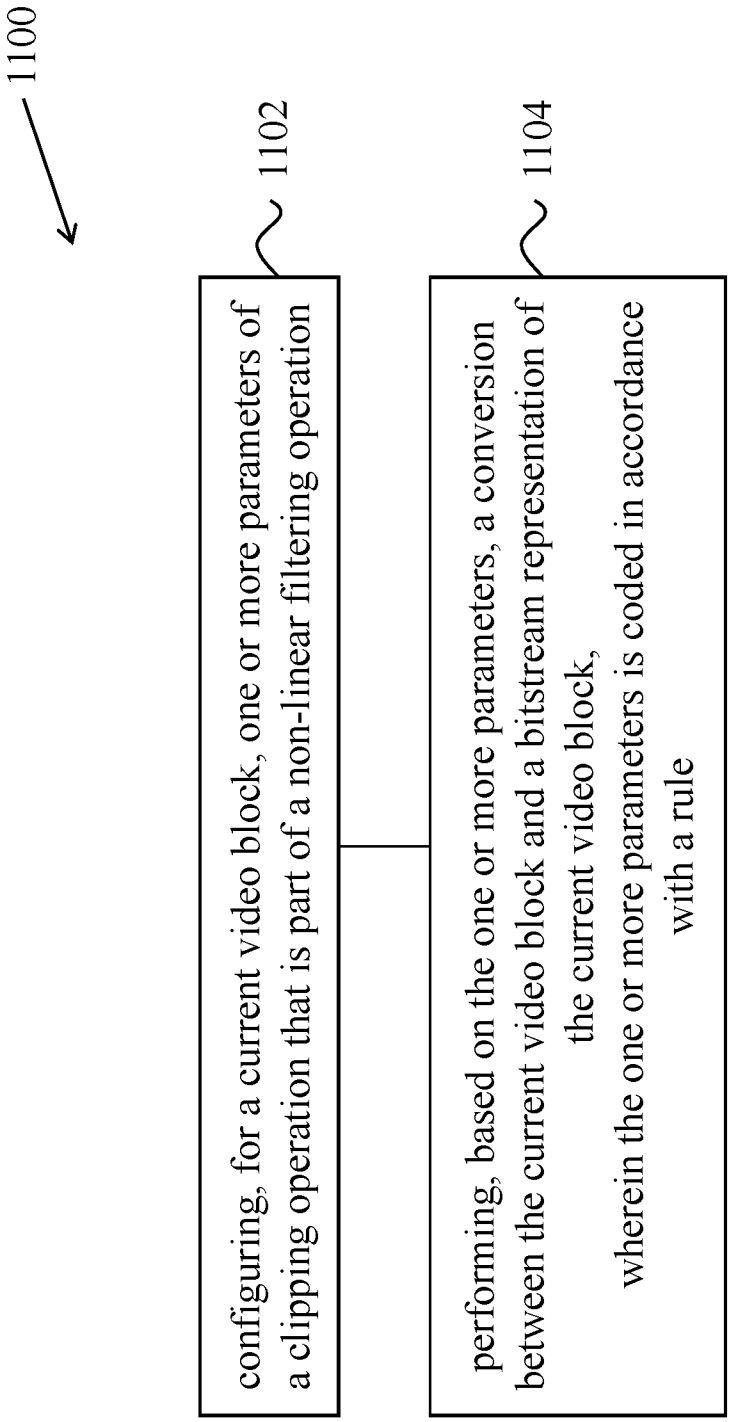
1100
1102
configuring, for a current video block, one or more parameters of a clipping operation that is part of a non-linear filtering operation
1104
performing, based on the one or more parameters, a conversion between the current video block and a bitstream representation of the current video block,
wherein the one or more parameters is coded in accordance with a rule

FIG. 11

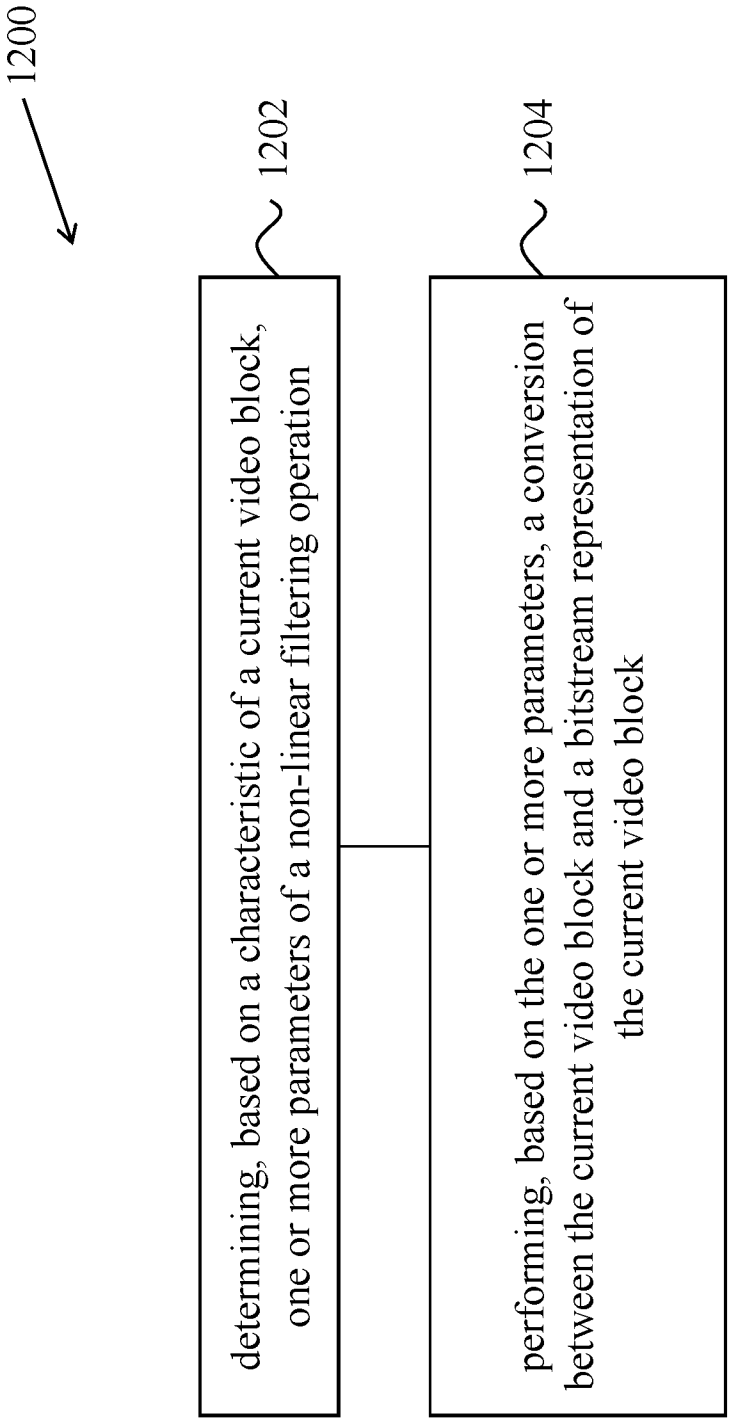
1200
1202
determining, based on a characteristic of a current video block, one or more parameters of a non-linear filtering operation
1204
performing, based on the one or more parameters, a conversion between the current video block and a bitstream representation of the current video block

FIG. 12

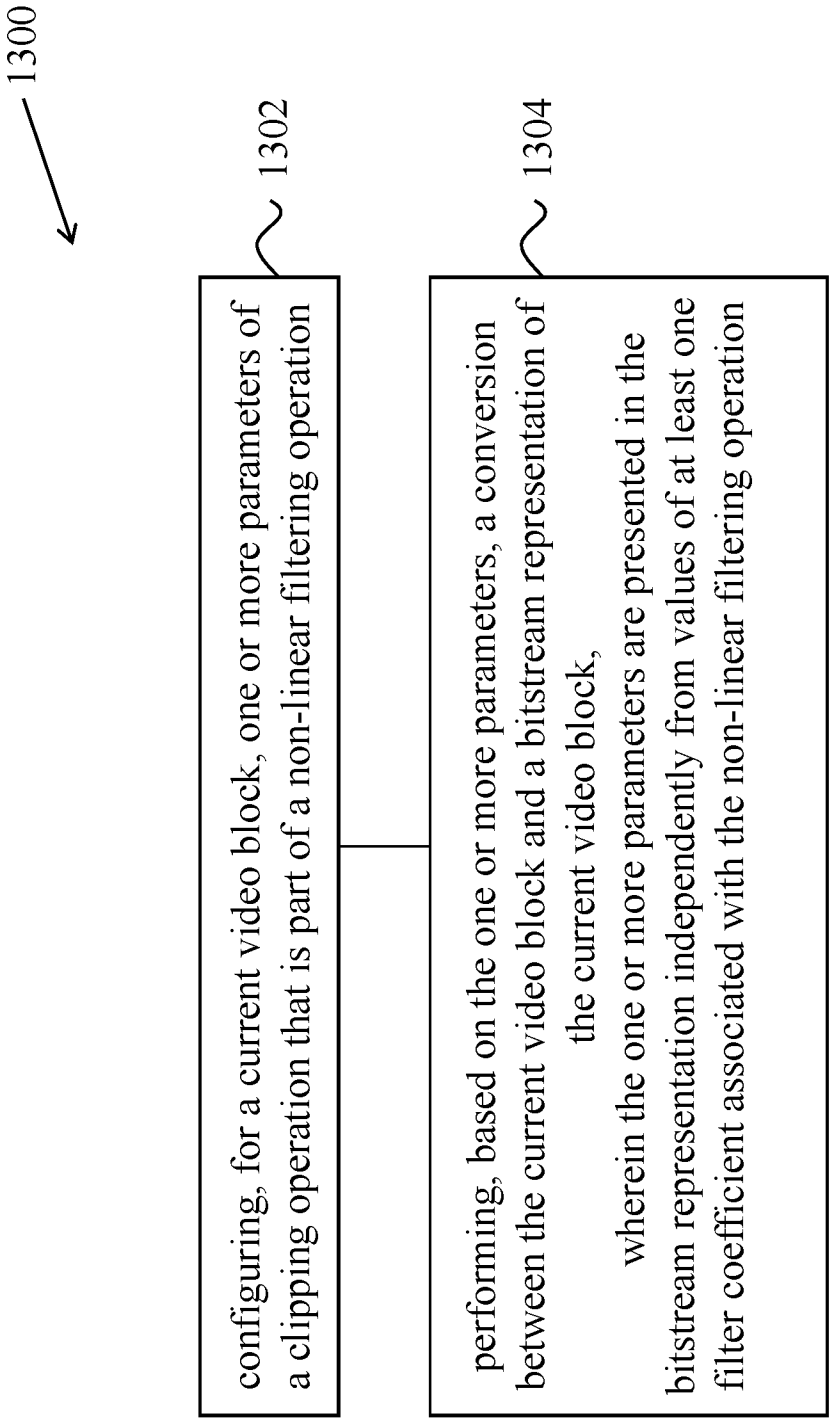
1300
1302
configuring, for a current video block, one or more parameters of a clipping operation that is part of a non-linear filtering operation
1304
performing, based on the one or more parameters, a conversion between the current video block and a bitstream representation of the current video block,
wherein the one or more parameters are presented in the bitstream representation independently from values of at least one filter coefficient associated with the non-linear filtering operation

FIG. 13

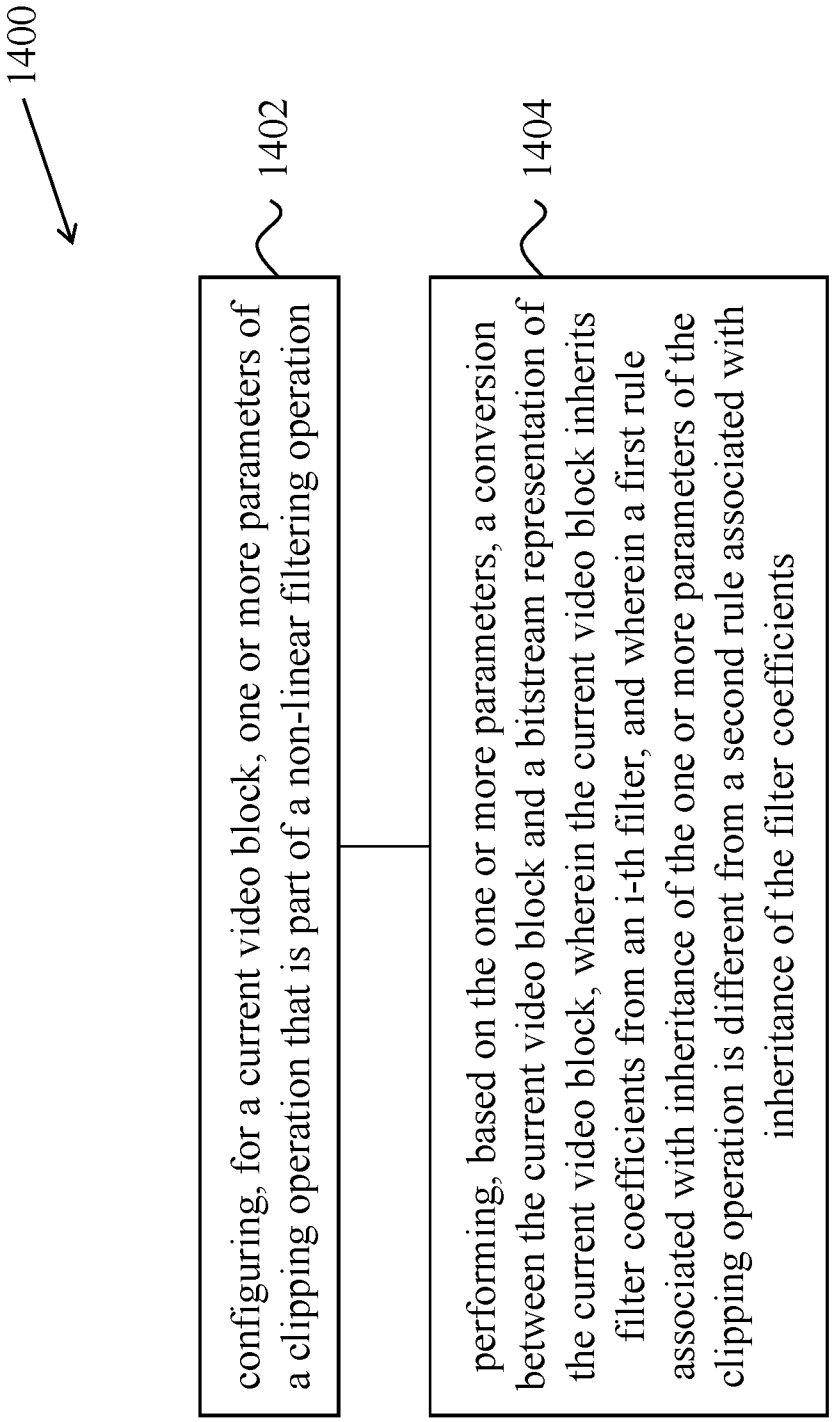
1400
1402
configuring, for a current video block, one or more parameters of a clipping operation that is part of a non-linear filtering operation
1404
performing, based on the one or more parameters, a conversion between the current video block and a bitstream representation of the current video block, wherein the current video block inherits filter coefficients from an i-th filter, and wherein a first rule associated with inheritance of the one or more parameters of the clipping operation is different from a second rule associated with inheritance of the filter coefficients

FIG. 14

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2020/084876

A. CLASSIFICATION OF SUBJECT MATTER

H04N 19/82(2014.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNPAT, CNKI, EPODOC, WPI: non-linear, adaptive, loop, filter, ALF, visual, media, video, parameter, configure, block, clipping, operation, coefficient, index, inheritance

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2011142136 A1 (HONG KONG APPLIED SCIENCE AND TECHNOLOGY RESEARCH INSTITUTE COMPANY LIMITED) 16 June 2011 (2011-06-16) abstract, description, paragraphs [0026]-[0056], figures 5-6	1-36
A	US 2016366422 A1 (DOLBY LABORATORIES LICENSING CORPORATION) 15 December 2016 (2016-12-15) the whole document	1-36
A	US 2016212002 A1 (DATANG MOBILE COMMUNICATIONS EQUIPMENT CO., LTD.) 21 July 2016 (2016-07-21) the whole document	1-36
A	US 2008239090 A1 (OHWAKI, Kazuyasu et al.) 02 October 2008 (2008-10-02) the whole document	1-36
A	CN 107909618 A (SHANGHAI UNITED IMAGING HEALTHCARE CO., LTD.) 13 April 2018 (2018-04-13) the whole document	1-36

☐ Further documents are listed in the continuation of Box C. ☑ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
09 June 2020	**30 June 2020**
Name and mailing address of the ISA/CN **National Intellectual Property Administration, PRC** **6, Xitucheng Rd., Jimen Bridge, Haidian District, Beijing 100088** **China** Facsimile No. **(86-10)62019451**	Authorized officer **JIANG,Lingling** Telephone No. **86-(10)-53961421**

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2020/084876

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
US	2011142136	A1	16 June 2011	CN	101841706	A	22 September 2010
US	2016366422	A1	15 December 2016	CN	110177276	A	27 August 2019
				CN	106063265	A	26 October 2016
				EP	3111645	A1	04 January 2017
				WO	2015130541	A1	03 September 2015
				JP	2017511045	A	13 April 2017
US	2016212002	A1	21 July 2016	EP	3029902	A1	08 June 2016
				CN	103491045	A	01 January 2014
				WO	2015032290	A1	12 March 2015
				JP	2016533687	A	27 October 2016
				KR	20160051868	A	11 May 2016
US	2008239090	A1	02 October 2008	JP	2008244591	A	09 October 2008
				EP	1975871	A2	01 October 2008
				CN	101276464	A	01 October 2008
CN	107909618	A	13 April 2018	CN	107886553	A	06 April 2018
				EP	3404617	A1	21 November 2018
				US	2018336678	A1	22 November 2018