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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
24 October 2013 (24.10.2013)

WIPO | PCT

(10) International Publication Number
WO 2013/155897 A1

(51) International Patent Classification:
H04N 7/26 (2006.01)

(21) International Application Number:
PCT/CN2013/071761

(22) International Filing Date:
22 February 2013 (22.02.2013)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/624,812 16 April 2012 (16.04.2012) US

(71) Applicant: **MEDIATEK INC.** [CN/CN]; No. 1 Dusing Road 1st, Science-Based Industrial Park, Hsin-Chu, Taiwan 300 (CN).

(72) Inventors: **FU, Chih-Ming**; No. 115, Aly.31, Ln.647, Sec.6, Zhonghua Rd., Xiangshan Dist., Hsinchu City, Taiwan 300 (CN). **TSAI, Chia-Yang**; 2F., No. 10, Aly.2, Ln.35, Minsheng St., Zhonghe Dist., New Taipei City, Taiwan 235 (CN). **HSU, Chih-Wei**; Rm.A, 8F., No. 2, Ln.83, Sec.1, Roosevelt Rd., Zhongzheng Dist., Taipei City, Taiwan 100 (CN). **CHEN, Ching-Yeh**; 4F., No. 809, Sec.5, Zhongxiao E. Rd., Nangang Dist., Taipei City, Taiwan 115 (CN). **HUANG, Yu-Wen**; 8F., No. 23, Ln. 298, Longjiang Rd., Zhongshan Dist., Taipei City, Taiwan 104 (CN). **LEI, Shaw-Min**; 6F., No. 32, Liujia 6th St., Zhubei City, Hsinchu County, Taiwan 302 (CN).

(74) Agent: **BEIJING SANYOU INTELLECTUAL PROPERTY AGENCY LTD.**; 16th Fl., Block A, Corporate Square, No. 35 Jinrong Street, Beijing 100033 (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, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, 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, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, 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, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, 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, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

[Continued on next page]

(54) Title: METHOD AND APPARATUS FOR LOOP FILTERING ACROSS SLICE OR TILE BOUNDARIES

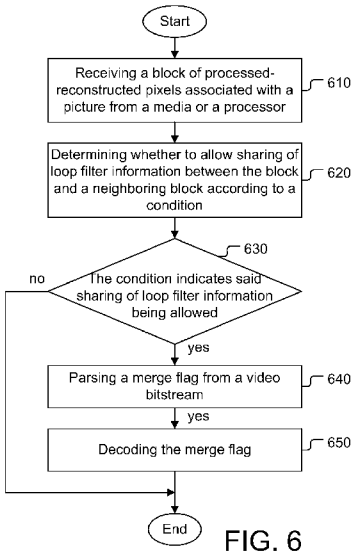


FIG. 6

(57) Abstract: A method and apparatus for loop filter processing of video data in a video encoder or decoder are disclosed. Embodiments according to the present invention conditionally allow sharing of loop filter parameters. In one embodiment, sharing of loop filter information between the current block and a neighboring block is determined according to a condition. If the condition indicates that sharing of loop filter information is allowed, a merge flag is coded and incorporated in the video bitstream in an encoder, and a merge flag is parsed from the video bitstream and decoded in a decoder.In one embodiment, the condition depends on region partitioning of the picture, where region partitioning partitions the picture into regions and the region may correspond to a slice or a tile. The condition is set to indicate that sharing of loop filter information is allowed if the block and the neighboring block are in a same slice/tile.

WO 2013/155897 A1

Published:

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

METHOD AND APPARATUS FOR LOOP FILTERING ACROSS SLICE OR TILE BOUNDARIES

BACKGROUND OF THE INVENTION

Cross Reference To Related Applications

[0001] The present invention claims priority to U.S. Provisional Patent Application, No. 61/624,812, filed on April 16, 2012, entitled "Prohibition against merging across slice/tile boundaries in SAO/ALF". The U.S. Provisional Patent Application is hereby incorporated by reference in its entirety.

Field of the Invention

[0002] The present invention relates to video coding. In particular, the present invention relates to video coding techniques associated with loop filtering and processing across slice or tile boundaries.

Description of the Related Art

[0003] Motion estimation is an effective inter-frame coding technique to exploit temporal redundancy in video sequences. Motion-compensated inter-frame coding has been widely used in various international video coding standards. The motion estimation adopted in various coding standards is often a block-based technique, where motion information such as coding mode and motion vector is determined for each macroblock or similar block configuration. In addition, intra-coding is also adaptively applied, where the picture is processed without reference to any other picture. The inter-predicted or intra-predicted residues are usually further processed by transformation, quantization, and entropy coding to generate a compressed video bitstream. During the encoding process, coding artifacts are introduced, particularly in the quantization process. In order to alleviate the coding artifacts, additional processing can be applied to reconstructed video to enhance picture quality in newer coding systems. The additional processing is often configured in an in-loop operation so that the encoder and the decoder may derive the same reference pictures.

[0004] Fig. 1 illustrates an exemplary adaptive inter/intra video coding system incorporating in-loop filtering process. For inter-prediction, Motion Estimation (ME)/Motion Compensation (MC) 112 is used to provide prediction data based on video data from other picture or pictures. Switch 114 selects Intra Prediction 110 or inter-prediction data from ME/MC 112 and the selected prediction data is supplied to Adder 116 to form *prediction errors*, also called

prediction residues or *residues*. The prediction error is then processed by Transformation (T) 118 followed by Quantization (Q) 120. The transformed and quantized residues are then coded by Entropy Encoder 122 to form a video bitstream corresponding to the compressed video data. The bitstream associated with the transform coefficients is then packed with side information such as motion, mode, and other information associated with the image unit. The side information may also be processed by entropy coding to reduce required bandwidth. Accordingly, the side information data is also provided to Entropy Encoder 122 as shown in Fig. 1 (the motion/mode paths to Entropy Encoder 122 are not shown). When the inter-prediction mode is used, a previously reconstructed reference picture or pictures have to be used to form prediction residues. Therefore, a reconstruction loop is used to generate reconstructed pictures at the encoder end. Consequently, the transformed and quantized residues are processed by Inverse Quantization (IQ) 124 and Inverse Transformation (IT) 126 to recover the processed residues. The processed residues are then added back to prediction data 136 by Reconstruction (REC) 128 to reconstruct the video data. The reconstructed video data may be stored in Reference Picture Buffer 134 and be used for prediction of other frames.

[0005] As shown in Fig. 1, incoming video data undergoes a series of processing in the encoding system. The reconstructed video data from REC 128 may be subject to various impairments due to the series of processing. Accordingly, various loop processing is applied to the reconstructed video data before the reconstructed video data is used as prediction data in order to improve video quality. In the High Efficiency Video Coding (HEVC) standard being developed, Deblocking Filter (DF) 130, Sample Adaptive Offset (SAO) 131 and Adaptive Loop Filter (ALF) 132 have been developed to enhance picture quality. The Deblocking Filter (DF) 130 is applied to boundary pixels and the DF processing is dependent on the underlying pixel data and coding information associated with the corresponding blocks. There is no DF-specific side information needs to be incorporated in the video bitstream. On the other hand, the SAO and ALF processing are adaptive, where filter information such as filter parameters and filter type may be dynamically changed according to the underlying video data. Therefore, filter information associated with SAO and ALF is incorporated in the video bitstream so that a decoder can properly recover the required information. Furthermore, filter information from SAO and ALF is provided to Entropy Encoder 122 for incorporation into the bitstream. In Fig. 1, DF 130 is applied to the reconstructed video first; SAO 131 is then applied to DF-processed video; and ALF 132 is applied to SAO-processed video. However, the processing order among DF, SAO and ALF may be re-arranged. In the High Efficiency Video Coding (HEVC) video standard being developed, the loop filtering process includes DF and SAO.

[0006] The coding process in HEVC is applied to each Largest Coding Unit (LCU). The LCU is adaptively partitioned into coding units using quadtree. Therefore, the LCU is also called coding tree block (CTB). In each leaf CU, DF is performed for each 8x8 block and in HEVC Test Model Version 5.0 (HM-5.0), the DF is applied to the 8x8 block boundaries. For each 8x8

block, horizontal filtering across vertical block boundaries is first applied, and then vertical filtering across horizontal block boundaries is applied.

[0007] Sample Adaptive Offset (SAO) 131 is also adopted in HM-5.0, as shown in Fig. 1. SAO is regarded as a special case of filtering where the processing only applies to one pixel. To apply SAO, a picture may be divided into multiple LCU-aligned regions. Each region can select one SAO type among two Band Offset (BO) types, four Edge Offset (EO) types, and no processing (OFF). For each to-be-processed (also called to-be-filtered) pixel, BO uses the pixel intensity to classify the pixel into a band. The pixel intensity range is equally divided into 32 bands, as shown in Fig. 2. Four consecutive bands are grouped together, where the starting band is indicated by sao_band_position. An exemplary 4-band group 200 is illustrated in Fig. 2. The first band position of this 4-band group is indicated by arrow 210. In EO, pixel classification is first done to classify pixels into different groups (also called categories or classes). The pixel classification for each pixel is based on a 3x3 window, as shown in Fig. 3 where four configurations corresponding to 0°, 90°, 135°, and 45° are used for classification. Upon classification of all pixels in a picture or a region, one offset is derived and transmitted for each group of pixels. In HM-5.0, SAO is applied to luma and chroma components, and each of the luma components is independently processed. Similar to BO, one offset is derived for all pixels of each category except for category 4 of EO, where Category 4 is forced to use zero offset. Table 1 below lists the EO pixel classification, where "C" denotes the pixel to be classified.

Table 1.

Category	Condition
0	C < two neighbors
1	C < one neighbor && C == one neighbor
2	C > one neighbor && C == one neighbor
3	C > two neighbors
4	None of the above

[0008] Adaptive Loop Filtering (ALF) 132 is another in-loop filtering in HM-5.0 to enhance picture quality, as shown in Fig. 1. Multiple types of luma filter footprints and chroma filter footprints are used. The ALF operation is applied in the horizontal direction first. After horizontal ALF is performed, ALF is applied in the vertical direction. In HM-5.0, up to sixteen luma ALF filters and at most one chroma ALF filter can be used for each picture. In order to allow localization of ALF, there are two modes for luma pixels to select filters. One is a Region-based Adaptation (RA) mode, and the other is a Block-based Adaptation (BA) mode. In addition to the RA and BA for adaptation mode selection at picture level, Coding Units (CUs) larger than a threshold can be further controlled by filter usage flags to enable or disable ALF operations locally. As for the chroma components, since they are relatively flat, no local adaptation is used in HM-5.0, and the two chroma components of a picture share the same filter. In MH-5.0, an ALF filter for a region may be selected from multiple ALF filters. In addition,

multiple filter footprints are used in HM-5.0. For each ALF filter, there is a set of filter coefficients associated with the filter. Therefore, the ALF information comprises identification for the selected ALF filter, the filter footprint and filter coefficients.

[0009] As shown in Fig. 1, DF 130 is applied to reconstructed pixels from REC 128. SAO 131 is then applied to DF-processed pixels and ALF 132 is applied to SAO-processed pixels. While the processing sequence illustrated in Fig. 1 is DF, SAO and ALF, other processing sequence may also be used. For example, SAO may be applied to reconstructed pixels from REC 128, DF-processed reconstructed pixels (i.e., DF applied to the reconstructed pixels), ALF-processed reconstructed pixels (i.e., ALF applied to reconstructed pixels), both DF-processed and ALF-processed pixels (i.e., DF applied to the reconstructed pixels and ALF applied to the DF-processed reconstructed pixels) or both ALF-processed and DF-processed pixels (i.e., ALF applied to the reconstructed pixels and DF applied to the ALF-processed reconstructed pixels). For convenience, the "*processed-reconstructed pixels*" may refer to any type of the processed pixels mentioned above during SAO processing. The "*processed-reconstructed pixels*" also includes the reconstructed pixels from REC 128. In this case, it can be considered that a null processing is applied to the reconstructed pixels from REC 128. Similarly, the "*processed-reconstructed pixels*" may also refer to various types of the processed pixels by DF, SAO, both DF and SAO or both SAO and DF during ALF processing. Again, for ALF processing, the "*processed-reconstructed pixels*" also includes the reconstructed pixels from REC 128.

[0010] To reduce side-information associated with SAO processing, SAO information of a current LCU can reuse the SAO information of a neighboring LCU above or to the left of the current LCU. The SAO information sharing is indicated by *merge* syntax. In HM-8.0, SAO syntax consists of sao_merge_left_flag, sao_merge_up_flag, sao_type_idx_luma, sao_type_index_chroma, sao_eo_class_luma, sao_eo_class_chroma, sao_band_position, sao_offset_abs, and sao_offset_sign, as shown in Table 2. Syntax sao_merge_left_flag indicates whether the current LCU reuses the SAO parameters of the left LCU. Syntax sao_merge_up_flag indicates whether the current LCU reuses the SAO parameters of the upper LCU. Syntax sao_type_idx represents the selected SAO type (sao_type_idx_luma and sao_type_idx_chroma for luma component and chroma component respectively). Syntax sao_offset_abs represents the offset magnitude and syntax sao_offset_sign represents the offset sign. Syntax cIdx indicates one of three color components. Similar mechanism can also be used to allow neighboring blocks to share the same ALF information.

Table 2.

sao(rx, ry){	**Descriptor**
if(rx > 0) {	
leftCtbInSliceSeg = CtbAddrInSliceSeg > 0	
leftCtbInTile = (TileId[CtbAddrInTS] = = TileId[CtbAddrRStoTS[CtbAddrInRS − 1]])	
if(leftCtbInSliceSeg && leftCtbInTile)	
sao_merge_left_flag	ae(v)
}	
if(ry > 0 && !sao_merge_left_flag) {	
upCtbInSliceSeg = (CtbAddrInRS − PicWidthInCtbsY) >= slice_segment_address	
upCtbInTile = (TileId[CtbAddrInTS] = = TileId[CtbAddrRStoTS[CtbAddrInRS − PicWidthInCtbsY]])	
if(upCtbInSliceSeg && upCtbInTile)	
sao_merge_up_flag	ae(v)
}	
if(!sao_merge_up_flag && !sao_merge_left_flag) {	
for(cIdx = 0; cIdx < 3; cIdx++) {	
if((slice_sao_luma_flag && cIdx = = 0) || (slice_sao_chroma_flag && cIdx > 0)) {	
if(cIdx = = 0)	
sao_type_idx_luma	ae(v)
else if(cIdx = = 1)	
sao_type_idx_chroma	ae(v)
if(SaoTypeIdx[cIdx][rx][ry] != 0) {	
for(i = 0; i < 4; i++)	
sao_offset_abs[cIdx][rx][ry][i]	ae(v)
if(SaoTypeIdx[cIdx][rx][ry] = = 1) {	
for(i = 0; i < 4; i++)	
if(sao_offset_abs[cIdx][rx][ry][i] != 0)	
sao_offset_sign[cIdx][rx][ry][i]	ae(v)
sao_band_position[cIdx][rx][ry]	ae(v)
} else {	
if(cIdx = = 0)	
sao_eo_class_luma	ae(v)
if(cIdx = = 1)	
sao_eo_class_chroma	ae(v)
}	
}	
}	
}	
}	
}	

[0011] The LCUs in a picture can be partitioned into slices, where each slice consists of multiple horizontally consecutive LCUs. In HM-5.0, another image unit structure, named *tile*, is introduced, where a picture is partitioned into multiple tiles. For example, a picture may be

divided into *M* tiles horizontally and *N* tiles vertically, where *M* and *N* are integers greater than 0. Each tile consists of multiple LCUs. Within each tile, the processing sequence of the LCUs is according to the raster scan order. Within each picture, the processing sequence of the tiles is also according to the raster scan order. Tile boundaries are often aligned with LCU boundaries.

[0012] In some systems, it is desirable to process the slices or tiles independently. Independent slice/tile processing will allow parallel processing of multiple slices or tiles. For CTBs or LCUs located at a left boundary or a top boundary of the slice or tile, SAO or ALF parameter sharing with a neighboring LCU above or to the left of the current LCU implies data dependency on an LCU from another slice or tile. Therefore, it is desirable to develop SAO or ALF processing that enables slice/tile independent processing.

BRIEF SUMMARY OF THE INVENTION

[0013] A method and apparatus for loop filter processing of video data in a video encoder or decoder are disclosed. Embodiments according to the present invention conditionally allow sharing of loop filter parameters. According to one embodiment of the present invention, sharing of loop filter information between the current block and a neighboring block is determined according to a condition. If the condition indicates that sharing of loop filter information is allowed, a merge flag is coded and incorporated in the video bitstream in an encoder or a merge flag is parsed from the video bitstream and decoded in a decoder. The merge flag may be a merge-left flag or a merge-up flag. The merge-left flag is used if the neighboring block is adjacent to the left side of the current block and the merge-up flag is used if the neighboring block is adjacent to the upper side of the current block.

[0014] One aspect of the invention addresses the condition regarding whether to allow sharing of loop filter information. In one embodiment, the condition depends on region partitioning of the picture, where region partitioning partitions the picture into regions. For example, the region may correspond to a slice or a tile. The condition is set to indicate that sharing of loop filter information is allowed if the current block and the neighboring block are in a same region. The condition is set to indicate that sharing of loop filter information is disallowed if the current block and the neighboring block are in different regions or if the neighboring block is not available. In another embodiment, the condition depends on whether the neighboring block is available. The condition is set to indicate that sharing of loop filter information is allowed if the neighboring block is available. On the other hand, the condition is set to indicate that sharing of loop filter information is disallowed if the neighboring block is not available. When the condition is set to indicate that sharing of loop filter information is disallowed, an encoder does not incorporate the merge flag in the video bitstream and a decoder does not parse the merge flag from the video bitstream.

[0015] The block may correspond to a coding tree block or a largest coding unit (LCU). The loop filter information may correspond to SAO (sample adaptive offset) information, ALF (adaptive loop filter) information, or DF (deblocking filter) information. The SAO information may comprise SAO type information, SAO offset values, edge offset type, band offset starting band position, offset magnitude, and offset sign. The ALF information may comprise ALF filter coefficients, ALF filter shape, filter selection index, filter classification method, and filter on/off control flag. Whether to allow sharing of loop filter information can be enabled or disabled based on a control flag. The control flag may be in slice header, picture parameter set, adaptation parameters set, sequence parameters set, or video parameter set of the video bitstream.

BRIEF DESCRIPTION OF DRAWINGS

[0016] Fig. 1 illustrates an exemplary video coding system using Inter/Intra prediction, where loop filter processing, including Deblocking Filter (DF), Sample Adaptive Offset (SAO) and Adaptive Loop Filter (ALF) is incorporated.

[0017] Fig. 2 illustrates an example of Band Offset (BO) by equally dividing the pixel intensity range into 32 bands.

[0018] Fig. 3 illustrates Edge Offset (EO) pixel classification based on a 3x3 window, with four configurations corresponding to 0°, 90°, 135°, and 45°.

[0019] Fig. 4A illustrates a vertical slice/tile boundary where two neighboring blocks belong to two different slices/tiles.

[0020] Fig. 4B illustrates a horizontal slice/tile boundary where two neighboring blocks belong to two different slices/tiles.

[0021] Fig. 5 illustrates an exemplary flowchart of sharing loop filter information conditionally for a video encoder incorporating an embodiment of the present invention.

[0022] Fig. 6 illustrates an exemplary flowchart of sharing loop filter information conditionally for a video decoder incorporating an embodiment of the present invention.

[0023] Fig. 7 illustrates an exemplary flowchart of coding SAO/ALF merge flag conditionally for a block in a slice/tile incorporating an embodiment of the present invention.

[0024] Fig. 8 illustrates an exemplary flowchart of parsing SAO/ALF merge flag conditionally for a block in a slice/tile incorporating an embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0025] In order to allow independent slice/tile processing for systems with loop filters such as SAO, ALF, and DF, embodiments according to the present invention conditionally allow

sharing of loop filter information. As mentioned before, the SAO processing incorporates merge syntax to allow SAO parameter sharing between neighboring LCUs. It is also possible to use similar syntax to allow ALF parameter sharing between neighboring LCUs. The present invention removes the data dependency associated with sharing of SAO, ALF, or DF information at slice/tile boundaries to allow independent slice/tile-based processing. If a block (a CTB or LCU) and an adjacent left block are within a same slice or tile, sharing of SAO, ALF, or DF information will be allowed. Similarly, if a block and an adjacent upper block are within a same slice or tile, sharing of SAO, ALF, or DF information will be allowed. When sharing of loop filter information is allowed, a merge flag (merge-left or merge-up) will be coded in an encoder according to whether the current block shares (or re-use) the loop filter information with the neighboring block. The coded merge flag will be incorporated in the video bitstream. In the decoder side, the merge flag will be parsed and used for decoding. If the block is located at the left-side boundary or the top-side boundary of a slice/tile, sharing of loop filter information with an adjacent left block or upper block respectively would require the loop filter information from another slice or tile. Therefore, sharing of loop filter information is disallowed in this case. When sharing of loop filter information is disallowed, the merge flag (either merge-left or merge-up) will not be used. Accordingly, an encoder will not code the merge flag nor incorporate the merge flag in the video bitstream. At the decoder side, there is no need to parse the merge flag. Accordingly, side information associated with the respective merge flag can be saved.

[0026] Examples of slice/tile independent processing are shown in Fig. 4A and Fig. 4B. Fig. 4A illustrates a vertical boundary 430 between tile 410 and tile 420. For block C (i.e., LCU C) at the left-side boundary of tile 420, no merge-left is allowed for loop filter information sharing because block C and its neighboring block A belong to different tiles. Fig. 4B illustrates an example of a horizontal slice/tile boundary 460 between slice/tile 440 and slice/tile 450. For block C at the top-side boundary of the slice/tile, no merge-up is allowed for loop filter information sharing because block C and its neighbor block B belong to different slices/tiles.

[0027] Fig. 5 illustrates an exemplary flowchart of sharing loop filter information conditionally for a video encoder incorporating an embodiment of the present invention. As shown in Fig. 1 and described in associated text, loop processing (DF, SAO or ALF) is always applied to reconstructed pixels. Furthermore, as mentioned before, the processed-reconstructed pixels may refer to various types of processed reconstructed pixels. A current block of processed-reconstructed pixels is received from a media or a processor as shown in step 510. The processed-reconstructed pixels may be stored in a media such as a RAM or DRAM in a system. Therefore, the processed-reconstructed pixels will have to be read back from a media. Also it is possible that the loop filter processing receives processed-reconstructed pixels directly from another processor (such as a central processing unit, a controller or a digital signal processor) responsible to generate the processed-reconstructed pixels. In this case, the processed-

reconstructed pixels will be received from a processor. The block can be a coding tree block (CTB), an LCU or other block unit (such as macroblock or other types). A decision is made in step 520 to determine whether to allow sharing of loop filter information between the current block and a neighboring block according to a condition. The condition is checked in step 530. If the condition indicates that sharing of loop filter information is allowed, a merge flag is coded (step 540) and the merge flag is incorporated in the video bitstream (step 550). Otherwise, steps 540 and 550 are skipped and no side information for the merge flag is incorporated in the video bit stream.

[0028] The neighboring block may not be available in some cases. For example, the current block may be located at the boundary of a picture. A respective neighboring block to the left or on the top of the current block may not exist since the respective neighboring block would be outside the picture. In this case, sharing of loop filter parameters will be disallowed. The loop filter information may correspond to SAO information, ALF information or DF information. The SAO information may comprise SAO type information, SAO offset values, edge offset type, band offset starting band position, offset magnitude, and offset sign. The ALF information may comprise ALF filter coefficients, ALF filter shape, filter selection index, filter classification method, and filter on/off control flag. In one embodiment, the condition depends on region partitioning of the picture, where the picture is partitioned into regions and the region may correspond to a slice or a tile. In slice/tile based loop-filter processing, the condition is set to indicate that sharing of loop filter information is allowed if the current block and the neighboring block are in a same slice/tile. On the other hand, the condition is set to indicate that sharing of loop filter information is disallowed if the block and the neighboring block are in different slices/tiles or if the neighboring block is not available. When sharing of loop filter information is disallowed, there is no need to code the merge flag or to incorporate the merge flag in the video bitstream. Accordingly, side information for the associated merge flags can be saved. This feature is useful even for non-slice/tile based loop filter processing. In another embodiment, the picture is not partitioned into regions and the condition depends on whether the neighboring block is available. The condition is set to indicate that sharing of loop filter information is allowed if the neighboring block is available. On the other hand, the condition is set to indicate that sharing of loop filter information is disallowed if the neighboring block is not available.

[0029] The conditional sharing of loop filter information between the current block and the neighboring block according to the condition can be enabled or disabled based on a control flag. The control flag can be incorporated in the slice header, picture parameter set, adaptation parameters set, sequence parameters set, or video parameter set of the video bitstream.

[0030] Fig. 6 illustrates an exemplary flowchart of sharing loop filter information conditionally for a video decoder incorporating an embodiment of the present invention. A current block of processed-reconstructed pixels is received from a media or a processor as shown in step 610. A

decision is made in step 620 to determine whether to allow sharing of loop filter information between the current block and a neighboring block according to a condition. The condition is checked in step 630. If the condition indicates that sharing of loop filter information is allowed, a merge flag is parsed from the video bitstream (step 640) and the merge flag is decoded (step 650). Otherwise, steps 640 and 650 are skipped.

[0031] In the case that the neighboring block is not available, sharing of loop filter parameters will be disallowed. In a decoder, the decision regarding whether sharing of loop filter information is allowed is the same as that for the encoder. When sharing of loop filter information is disallowed, there is no need to parse the merge flag from the video bitstream or to decode the merge flag.

[0032] The conditional sharing of loop filter information between the current block and the neighboring block according to the condition can be enabled or disabled based on a control flag. The control flag can be parsed from the slice header, picture parameter set, adaptation parameters set, sequence parameters set, or video parameter set of the video bitstream.

[0033] A flowchart for slice/tile based SAO/ALF processing at the encoder side incorporating an embodiment of the present invention is shown in Fig. 7. A decision is made in step 710 to determine whether the current block and its neighboring block adjacent to the left side of the current block are in the same tile/slice. If the current block and the neighboring block adjacent to the left side of the current block are within the same slice/tile, the coding process for the SAO/ALF merge-left flag is performed as shown in step 720 and the coded merge-left flag is then incorporated in the video bitstream as shown in step 730. Otherwise, steps 720 and 730 are skipped and no side information for the merge-left flag is needed. When the neighboring block adjacent to the left side of the current block is not available (for example, the current block being a boundary block at the left side of a slice/tileand the slice/tile is located at the left side boundary of a picture), the decision in step 710 will also take the “no” branch. In step 720, the SAO/ALF merge-left flag is coded according to whether the two respective blocks share loop filter information. The value of the SAO/ALF merge-left flag is indicative of whether the current block shares SAO/ALF parameters with the left neighboring block.

[0034] Similar process is applied to determine whether to allow merge-up flag for a current block to share loop filter information with a neighboring block adjacent to the top side of the current block as shown in step 740. If the current block and the neighboring block adjacent to the top side of the current block are in the same tile/slice, sharing of loop filter information is allowed. In this case, the coding process for the SAO/ALF merge-up flag is performed as shown in step 750 and the merge-up flag coded is then incorporated in the video bitstream as shown in step 760. Otherwise, steps 750 and 760 are skipped and no side information for the merge-up flag is needed. When the neighboring block adjacent to the top side of the current block is not available (for example, the current block being a boundary block at the top side of a slice/tile and the slice/tile is located at the top side boundary of a picture), the decision in step

740 will also take the "no" branch. In step 740, if the current block is already merged to other block (i.e. left block), no merge-up processing or flag is needed. In step 750, the SAO/ALF merge-up flag is coded according to whether the two respective blocks share loop filter information. The value of the SAO/ALF merge-up flag is indicative of whether the current block shares SAO/ALF parameters with the upper neighboring block.

[0035] The exemplary flowchart shown in Fig. 7 is for illustration purpose. A skilled person in the art may re-arrange, combine steps or split a step to practice the present invention without departing from the spirit of the present invention. For example, the merge-up flag can be checked first and then the merge-left flag. In this case, steps 740 through 760 are performed before steps 710 through 730. In another example, the decision process for merge-left flag and merge-up flag can be performed first. After both merge-left flag and merge-up flag are determined, the two flags are coded and incorporated in the video bitstream.

[0036] A flowchart for slice/tile based SAO/ALF processing at the decoder side incorporating an embodiment of the present invention is shown in Fig. 8. A decision is made in step 810 to determine whether the current block and its neighboring block adjacent to the left side of the current block are in the same tile/slice. If the current block and the neighboring block adjacent to the left side of the current block are within the same slice/tile, the parsing process to extract the SAO/ALF merge-left flag is performed as shown in step 820 and the merge-left flag is decoded in step 830. Otherwise, steps 820 and 830 are skipped. When the neighboring block adjacent to the left side of the current block is not available, the decision in step 810 will also take the "no" branch. Similar process is applied to check whether to allow merge-up flag for a current block to share loop filter information with a neighboring block adjacent to the top side of the current block as shown in step 840. In step 840, if the current block is already merged to other block (i.e. left block), no merge-up flag needs to be parsed and decoded. If the current block and the neighboring block adjacent to the top side of the current block are in the same tile/slice, sharing of loop filter information is allowed. In this case, the parsing process to extract the SAO/ALF merge-up flag is performed as shown in step 850 and the merge-up flag is decoded in step 860. Otherwise, steps 850 and 860 are skipped. When the neighboring block adjacent to the top side of the current block is not available, the decision in step 840 will also take the "no" branch.

[0037] The above description is presented to enable a person of ordinary skill in the art to practice the present invention as provided in the context of a particular application and its requirement. Various modifications to the described embodiments will be apparent to those with skill in the art, and the general principles defined herein may be applied to other embodiments. Therefore, the present invention is not intended to be limited to the particular embodiments shown and described, but is to be accorded the widest scope consistent with the principles and novel features herein disclosed. In the above detailed description, various specific details are illustrated in order to provide a thorough understanding of the present

invention. Nevertheless, it will be understood by those skilled in the art that the present invention may be practiced.

[0038] Embodiment of the present invention as described above may be implemented in various hardware, software codes, or a combination of both. For example, an embodiment of the present invention can be a circuit integrated into a video compression chip or program code integrated into video compression software to perform the processing described herein. An embodiment of the present invention may also be program code to be executed on a Digital Signal Processor (DSP) to perform the processing described herein. The invention may also involve a number of functions to be performed by a computer processor, a digital signal processor, a microprocessor, or field programmable gate array (FPGA). These processors can be configured to perform particular tasks according to the invention, by executing machine-readable software code or firmware code that defines the particular methods embodied by the invention. The software code or firmware code may be developed in different programming languages and different formats or styles. The software code may also be compiled for different target platforms. However, different code formats, styles and languages of software codes and other means of configuring code to perform the tasks in accordance with the invention will not depart from the spirit and scope of the invention.

[0039] The invention may be embodied in other specific forms without departing from its spirit or essential characteristics. The described examples are to be considered in all respects only as illustrative and not restrictive. The scope of the invention is therefore, indicated by the appended claims rather than by the foregoing description. All changes which come within the meaning and range of equivalency of the claims are to be embraced within their scope.

CLAIMS

1、A method for loop filter processing of video data in a video decoder, the method comprising:

receiving a block of processed-reconstructed pixels associated with a picture from a media or a processor;

determining whether to allow sharing of loop filter information between the block and a neighboring block according to a condition, wherein the condition depends on region partitioning of the picture;

parsing a merge flag from a video bitstream if the condition indicates said sharing of loop filter information being allowed; and

decoding the merge flag if the condition indicates said sharing of loop filter information being allowed.

2、The method of Claim 1, wherein the merge flag consists of a merge-left flag and a merge-up flag.

3、The method of Claim 2, wherein the merge-left flag is used if the neighboring block is adjacent to left side of the block.

4、The method of Claim 2, wherein the merge-up flag is used if the neighboring block is adjacent to upper side of the block.

5、The method of Claim 2, wherein the merge-up flag is used if merge-left flag indicated said sharing of loop filter information being not allowed.

6、The method of Claim 1, wherein said region partitioning partitions the picture into regions and the region corresponds to a slice or a tile.

7、The method of Claim 1, wherein the condition is set to indicate said sharing of loop filter information being disallowed if the block and the neighboring block are in different regions or if the neighboring block is not available.

8、The method of Claim 1, wherein the block corresponds to a coding tree block or a largest coding unit (LCU).

9、The method of Claim 1, wherein the loop filter information corresponds to SAO (sample adaptive offset) information, ALF (adaptive loop filter) information, or DF (deblocking filter) information.

10、The method of Claim 9, wherein the SAO information comprises SAO type information, SAO offset values, edge offset type, band offset starting band position, offset magnitude, and offset sign.

11、The method of Claim 9, wherein the ALF information comprises filter coefficients, and filter shape, filter selection index, filter classification method, and filter on/off control flag.

12、The method of Claim 1, wherein said determining whether to allow sharing of loop filter information between the block and the neighboring block according to the condition is enabled or disabled based on a control flag.

13、The method of Claim 12, wherein the control flag is parsed from slice header, picture parameter set, adaptation parameters set, sequence parameters set, or video parameter set of the video bitstream.

14、A method for loop filter processing of video data in a video encoder, the method comprising:

receiving a block of processed-reconstructed pixels associated with a picture from a media or a processor;

determining whether to allow sharing of loop filter information between the block and a neighboring block according to a condition, wherein the condition depends on region partitioning of the picture;

coding a merge flag if the condition indicates said sharing of loop filter information being allowed; and

incorporating the merge flag in a video bitstream if the condition indicates said sharing of loop filter information being allowed.

15、The method of Claim14, wherein the merge flag consists of a merge-left flag and a merge-up flag.

16、The method of Claim 15, wherein the merge-up flag is used if merge-left flag indicated said sharing of loop filter information being not allowed.

17、The method of Claim 14, wherein said region partitioning partitions the picture into regions and the region corresponds to a slice or a tile.

18、The method of Claim 14, wherein the condition is set to indicate said sharing of loop filter information being disallowed if the block and the neighboring block are in different regions or if the neighboring block is not available.

19、The method of Claim 14, wherein the block corresponds to a coding tree block or a largest coding unit (LCU).

20、The method of Claim 14, wherein the loop filter information corresponds to SAO (sample adaptive offset) information, ALF (adaptive loop filter) information, or DF (deblocking filter) information.

21、The method of Claim 14, wherein said determining whether to allow sharing of loop filter information between the block and the neighboring block according to the condition is enabled or disabled based on a control flag.

22、The method of Claim21, wherein the control flag is incorporated in slice header, picture parameter set, adaptation parameters set, sequence parameters set, or video parameter set

of the video bitstream.

23、An apparatus for loop filter processing of video data in a video decoder, the apparatus comprising:

means for receiving a block of processed-reconstructed pixels associated with a picture from a media or a processor;

means for determining whether to allow sharing of loop filter information between the block and a neighboring block according to a condition, wherein the condition depends on region partitioning of the picture;

means for parsing a merge flag from a video bitstream if the condition indicates said sharing of loop filter information being allowed; and

means for decoding the merge flag if the condition indicates said sharing of loop filter information being allowed.

24、An apparatus for loop filter processing of video data in a video encoder, the apparatus comprising:

means for receiving a block of processed-reconstructed pixels associated with a picture from a media or a processor;

means for determining whether to allow sharing of loop filter information between the block and a neighboring block according to a condition, wherein the condition depends on region partitioning of the picture;

means for coding a merge flag if the condition indicates said sharing of loop filter information being allowed; and

means for incorporating the merge flag in a video bitstream if the condition indicates said sharing of loop filter information being allowed.

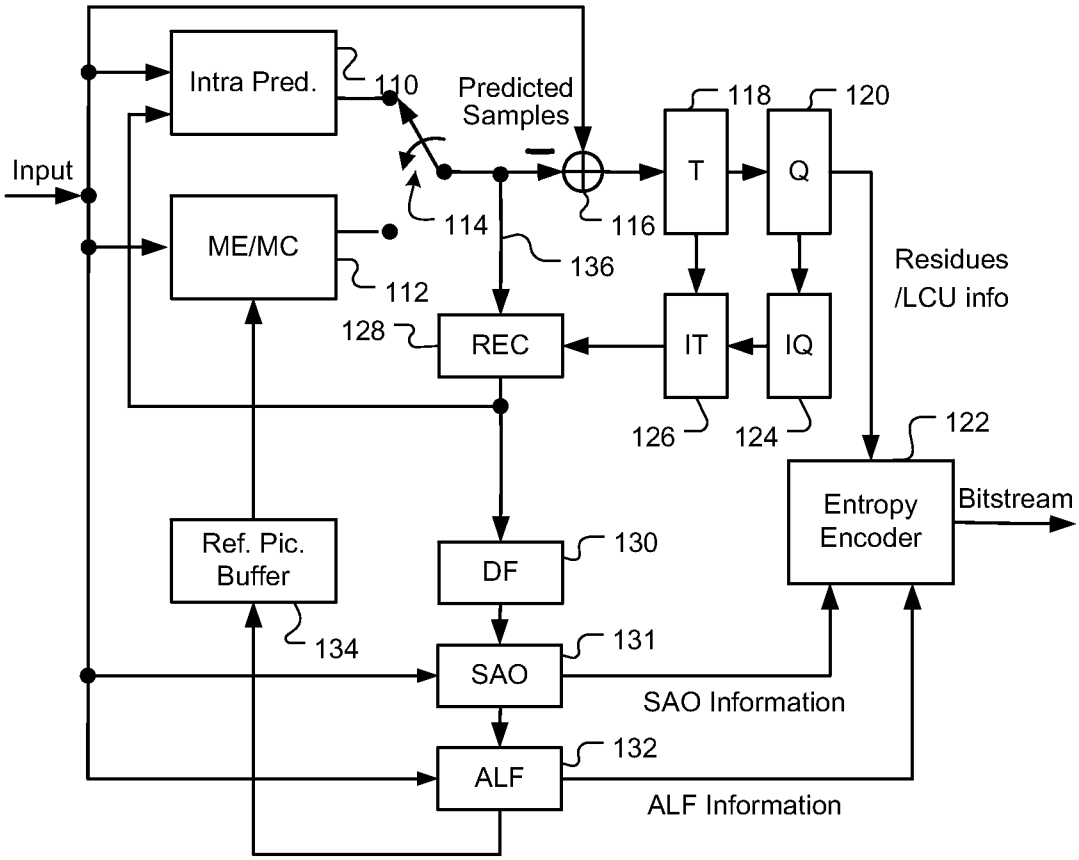

Intra Pred.
110
Predicted Samples
118
120
Input
T
Q
ME/MC
114
116
136
Residues /LCU info
112
128
REC
IT
IQ
126
124
122
Entropy Encoder
Bitstream
Ref. Pic. Buffer
DF
130
134
131
SAO
SAO Information
132
ALF
ALF Information

FIG. 1

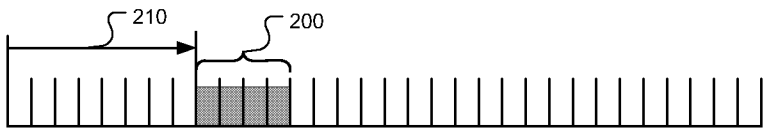

210
200

FIG. 2

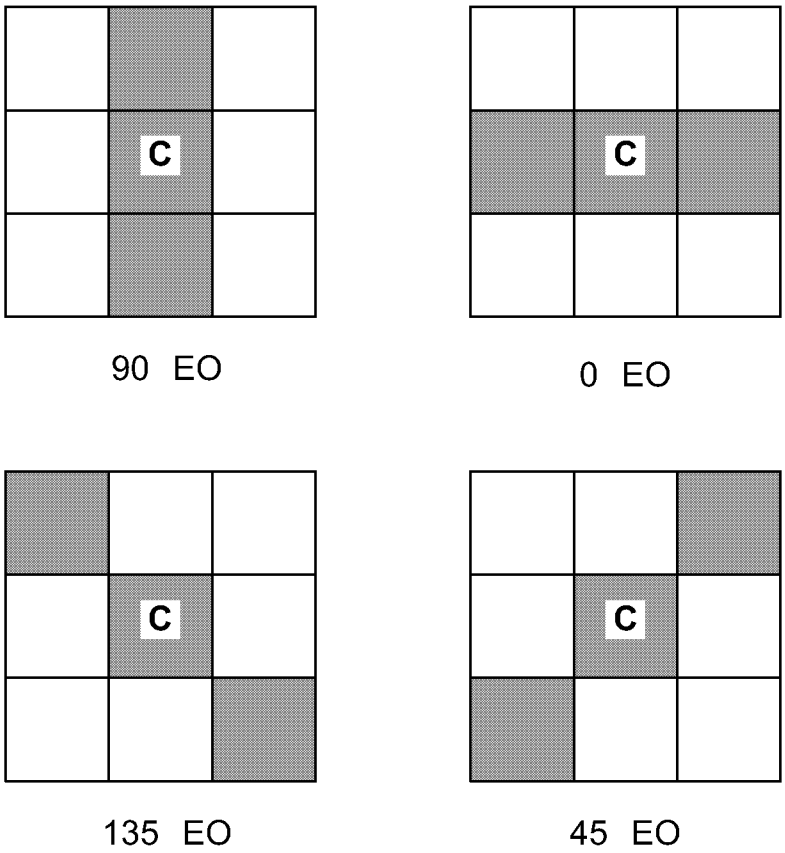


FIG. 3

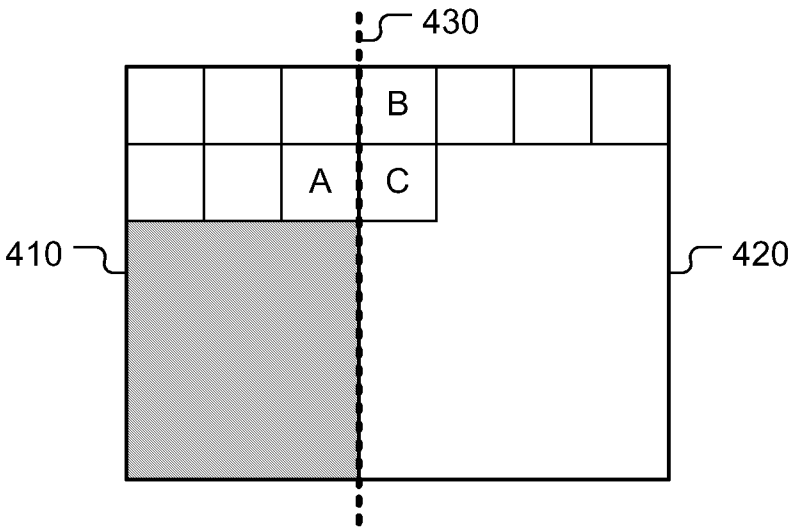


FIG. 4A

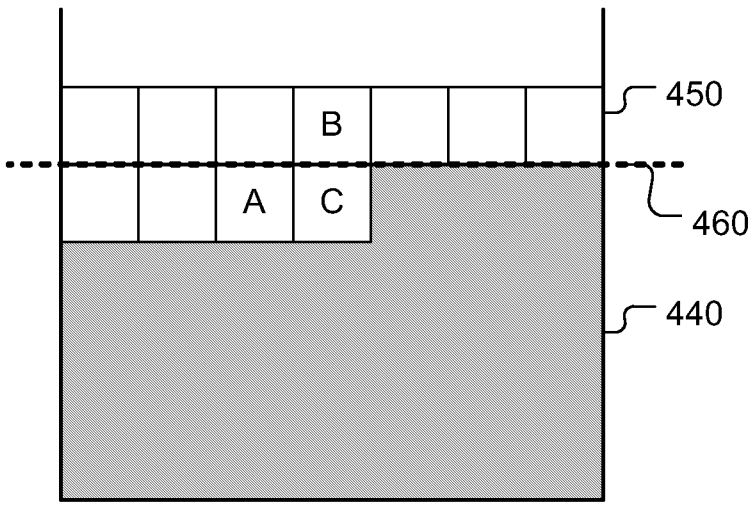


FIG. 4B

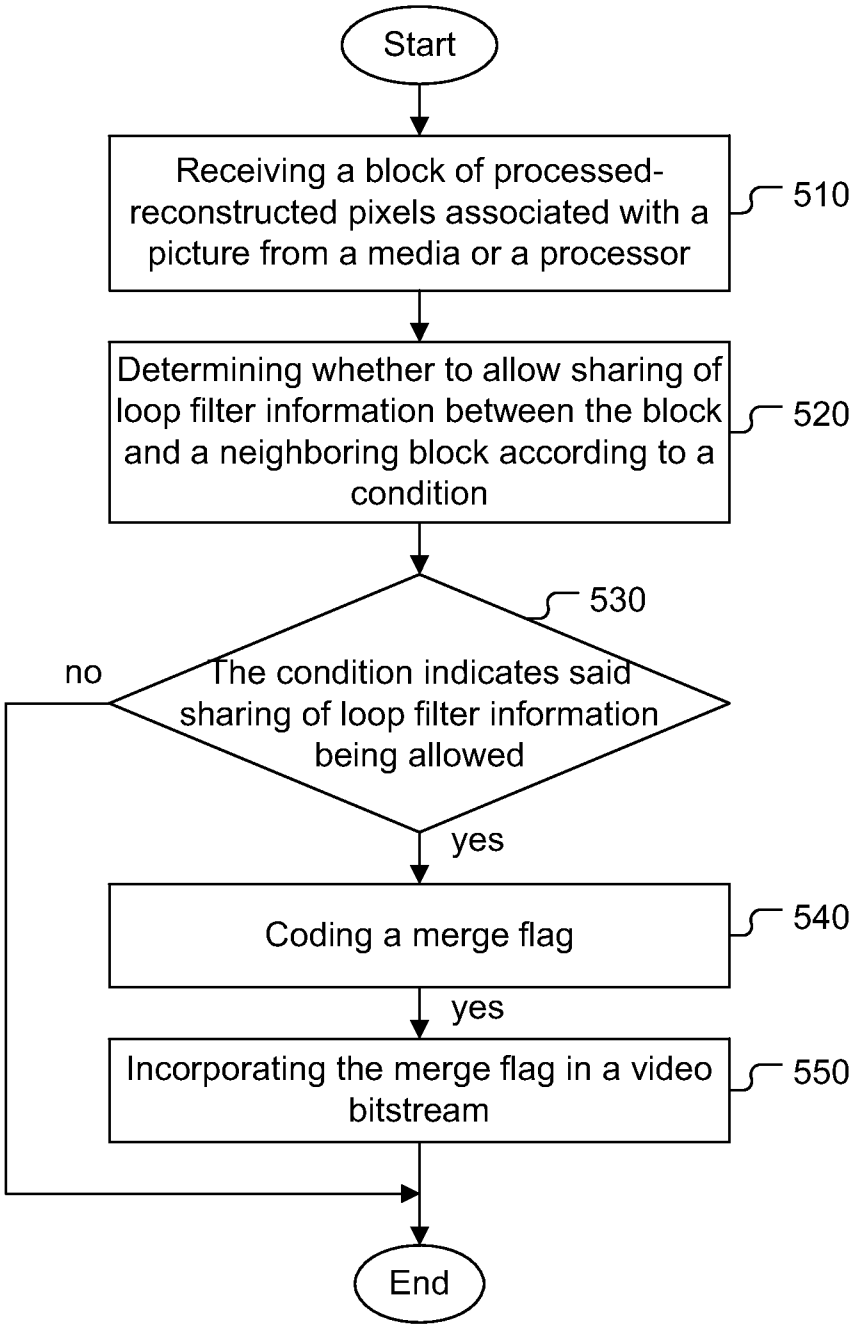

Start
Receiving a block of processed-reconstructed pixels associated with a picture from a media or a processor
510
Determining whether to allow sharing of loop filter information between the block and a neighboring block according to a condition
520
530
The condition indicates said sharing of loop filter information being allowed
no
yes
Coding a merge flag
540
yes
Incorporating the merge flag in a video bitstream
550
End

FIG. 5

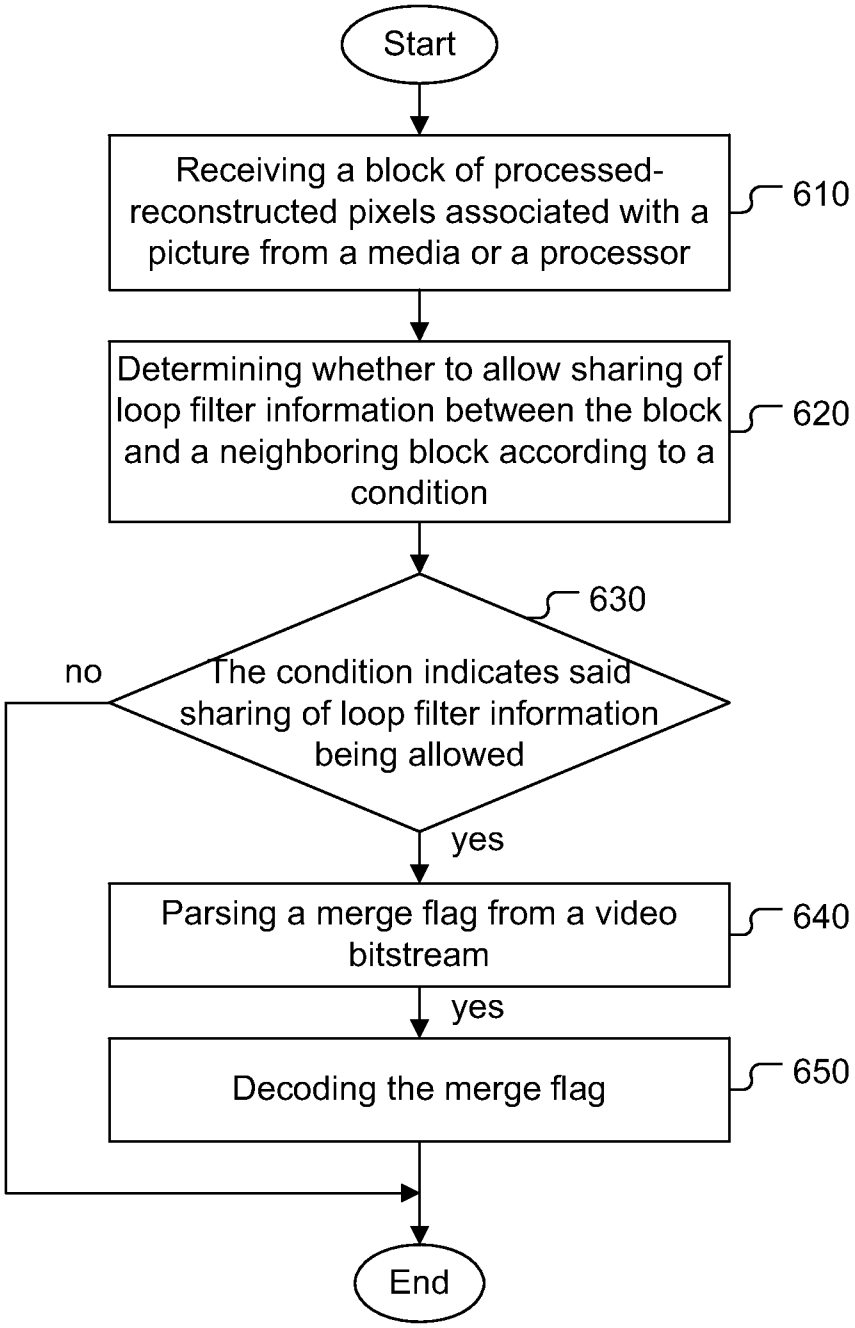

Start
Receiving a block of processed-reconstructed pixels associated with a picture from a media or a processor
610
Determining whether to allow sharing of loop filter information between the block and a neighboring block according to a condition
620
630
The condition indicates said sharing of loop filter information being allowed
no
yes
Parsing a merge flag from a video bitstream
640
yes
Decoding the merge flag
650
End

FIG. 6

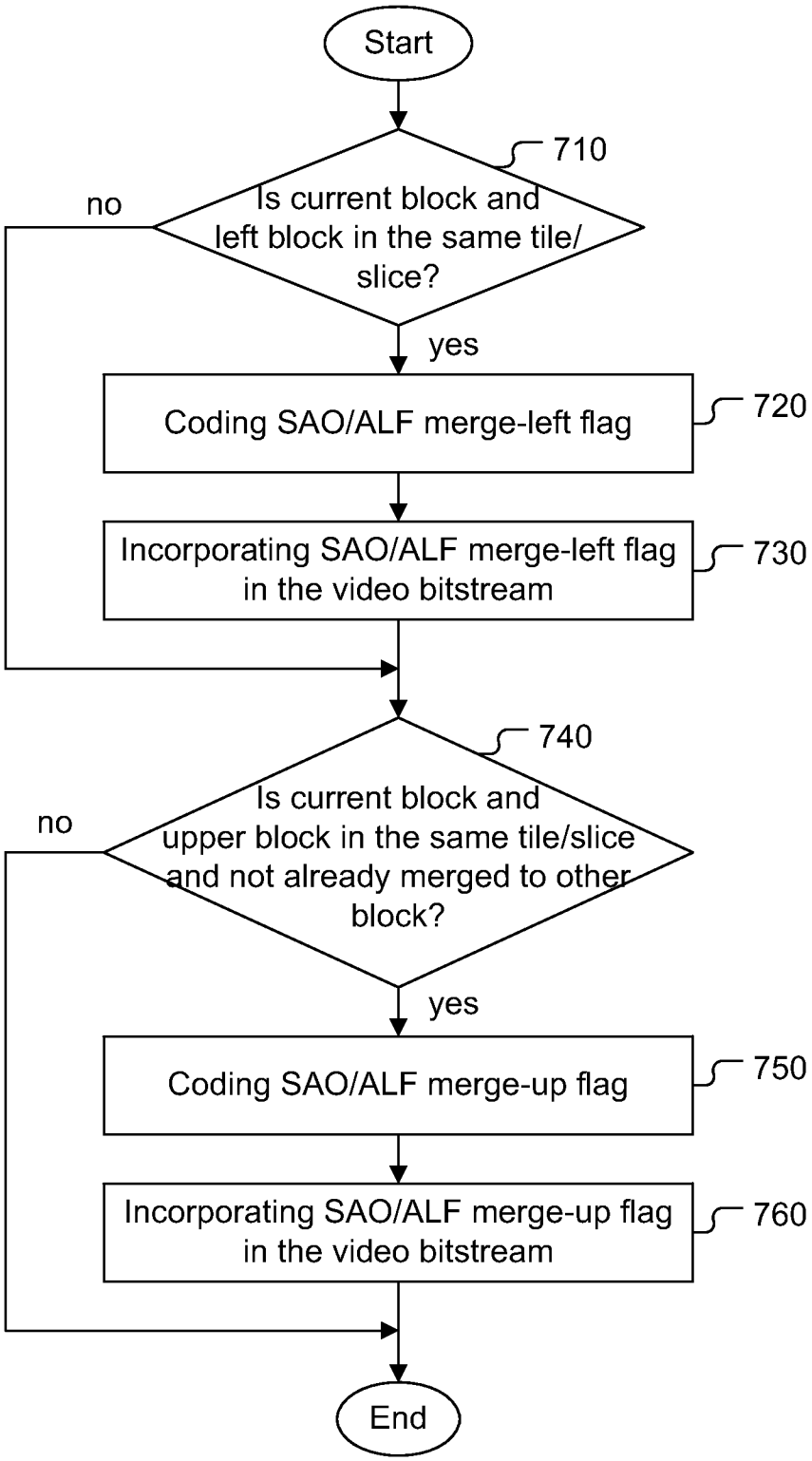

Start
710
Is current block and left block in the same tile/ slice?
no
yes
Coding SAO/ALF merge-left flag
720
Incorporating SAO/ALF merge-left flag in the video bitstream
730
740
Is current block and upper block in the same tile/slice and not already merged to other block?
no
yes
Coding SAO/ALF merge-up flag
750
Incorporating SAO/ALF merge-up flag in the video bitstream
760
End

FIG. 7

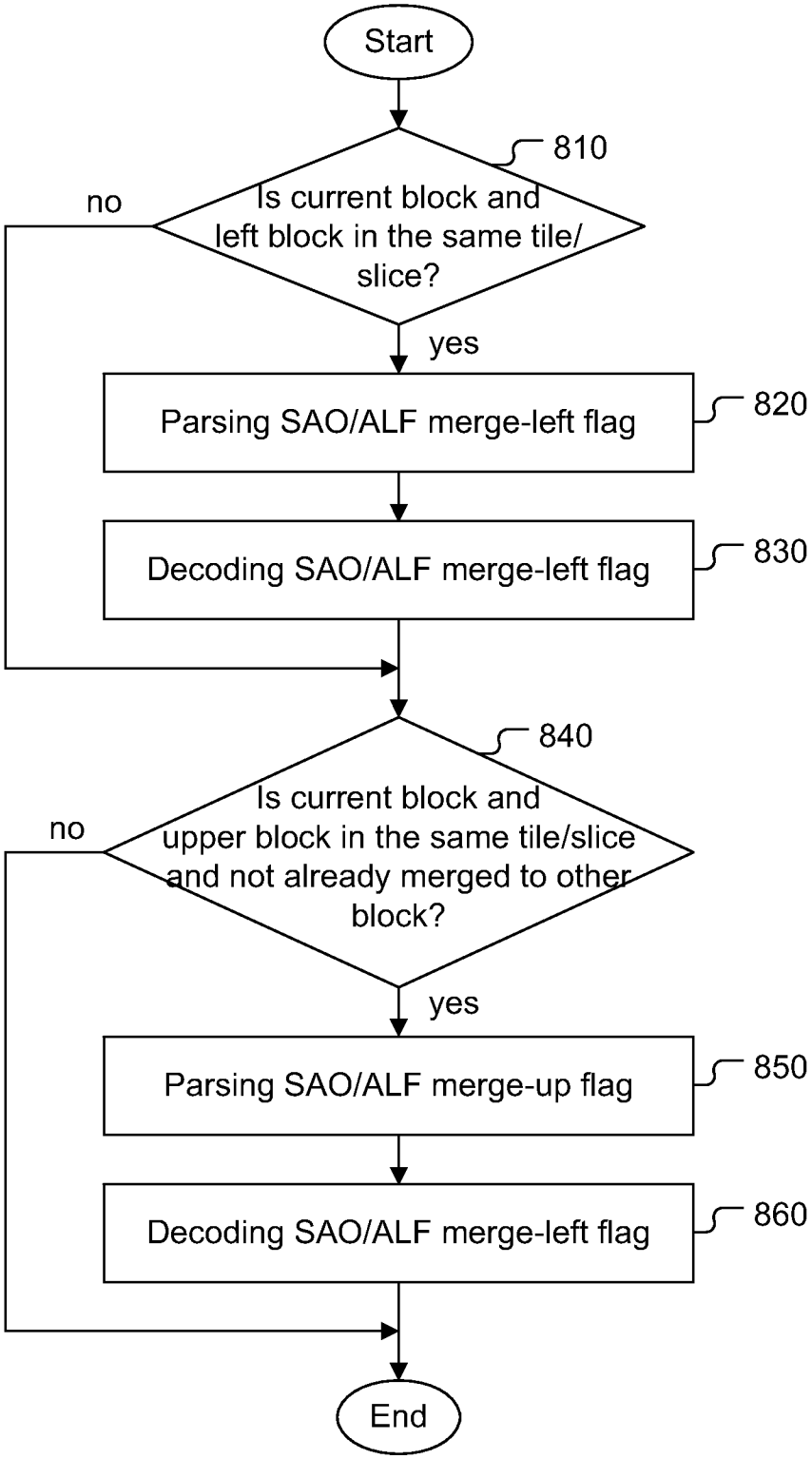

Start
810
Is current block and left block in the same tile/ slice?
no
yes
Parsing SAO/ALF merge-left flag
820
Decoding SAO/ALF merge-left flag
830
840
Is current block and upper block in the same tile/slice and not already merged to other block?
no
yes
Parsing SAO/ALF merge-up flag
850
Decoding SAO/ALF merge-left flag
860
End

FIG. 8

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2013/071761

A. CLASSIFICATION OF SUBJECT MATTER

H04N7/26(2006.01)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)

IPC: H04N, H04B

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)

WPI,EPODOC,CNPAT,CNKI: loop, filter, decoder, encoder, sharing, share, merge, flag, reconstruct+, block, pixel, region, partition, parse, incorporate, neighboring, slice, tile

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US2011/0274158A1 (FU, Chih-Ming et al.) 10 Nov. 2011 (10.11.2011) claims 1-9, 14-20, description paragraphs 0026-0043	1-24
A	EP1944974A1 (MATSUSHITA ELECTRIC INDUSTRIAL CO., LTD.) 16 Jul. 2008 (16.07.2008) the whole document	1-24
A	CN101517909A (FREESCALE SEMICONDUCTOR INC.) 26 Aug. 2009 (26.08.2009) the whole document	1-24
A	CN102150428A (GOOGLE INC.) 10 Aug. 2011 (10.08.2011) the whole document	1-24

☐ 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
04 May 2013 (04.05.2013)	**23 May 2013 (23.05.2013)**
Name and mailing address of the ISA/CN The State Intellectual Property Office, the P.R.China 6 Xitucheng Rd., Jimen Bridge, Haidian District, Beijing, China 100088 Facsimile No. 86-10-62019451	Authorized officer CI,Xue Telephone No. (86-10)62413418

Form PCT/ISA /210 (second sheet) (July 2009)

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2013/071761

Patent Documents referred in the Report	Publication Date	Patent Family	Publication Date
US2011/0274158A1	10.11.2011	WO2011140960 A1	17.11.2011
		WO2012045270 A1	12.04.2012
		DE112011100420T5	13.12.2012
		CN102804776A	28.11.2012
EP1944974A1	16.07.2008	WO2008084745 A1	17.07.2008
		EP2104355 A1	23.09.2009
		US2010021071 A1	28.01.2010
		JP2008553083T2	30.04.2010
CN101517909A	26.08.2009	US2008069247 A1	20.03.2008
		WO2008033595 A2	20.03.2008
		TW200818922 A	16.04.2008
		EP2067265 A2	10.06.2009
		KR20090064388 A	18.06.2009
		JP2010504036 A	04.02.2010
		INDELNP200900650E	20.08.2010
CN102150428A	10.08.2011	US2010061444 A1	11.03.2010
		WO2010030761 A2	18.03.2010
		CA2736890 A1	18.03.2010
		EP2324637 A2	25.05.2011
		KR20110056388 A	27.05.2011
		JP2012502593 A	26.01.2012