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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
04 February 2021 (04.02.2021)

WIPO | PCT

(10) International Publication Number
WO 2021/021698 A1

(51) International Patent Classification:
H04N 19/573 (2014.01) *H04N 19/122* (2014.01)
H04N 19/513 (2014.01) *H04N 19/176* (2014.01)
H04N 19/119 (2014.01) *H04N 19/44* (2014.01)

(21) International Application Number:
PCT/US2020/043681

(22) International Filing Date:
27 July 2020 (27.07.2020)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/879,459 27 July 2019 (27.07.2019) US

(71) Applicant: **BEIJING DAJIA INTERNET INFORMATION TECHNOLOGY CO., LTD.** [CN/CN]; Room 101D1-7, 1st Floor, Building 1, No. 6, Shangdi West Road, Haidian District, Beijing, Beijing 100085 (CN).

(72) Inventor; and

(71) Applicant *(for US only)*: **CHEN, Yi-Wen** [CN/US]; 8910 University Center Lane, Suite 400, San Diego, California 92122 (US).

(72) Inventors: **XIU, Xiaoyu**; 6691 Aliso Valley Way, San Diego, California 92130 (US). **WANG, Xianglin**; 8910 University Center Lane, Suite 400, San Diego, California 92122 (US). **YU, Bing**; Room 101D1-7, 1st Floor, Building 1, No. 6, Shangdi West Road, Haidian District, Beijing, Beijing 100092 (CN).

(74) Agent: **HAO TAN**; Arch & Lake LLP, 203 N. LaSalle Street, Suite 2100, Chicago, Illinois 60601 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM,

(54) Title: METHODS AND APPARATUSES FOR DECODER-SIDE MOTION VECTOR REFINEMENT IN VIDEO CODING

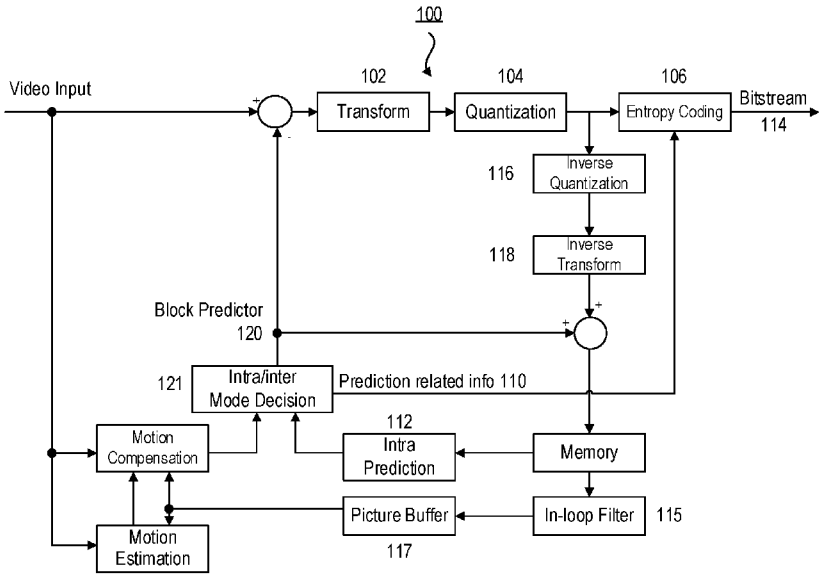

FIG. 1

(57) Abstract: A method for video coding is provided. The method includes: partitioning a video picture into a plurality of coding units (CUs), wherein each of the plurality of CUs comprises a plurality of sub-blocks; deriving an initial motion vector (MV) for each of the plurality of sub-blocks; deriving a plurality of motion vector (MV) candidates for bi-directional optical flow (BDOF) and decoder-side motion vector refinement (DMVR); determining an original cost value for the initial MV; determining an updated cost value for the initial MV; obtaining a first threshold and a second threshold for each of the plurality of sub-blocks; determining whether to terminate bi-directional optical flow (BDOF) for each of the plurality of sub-blocks based on the first threshold; and determining whether to terminate decoder-side motion vector refinement (DMVR) for each of the plurality of sub-blocks based on the second threshold.

WO 2021/021698 A1

[Continued on next page]

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, IT, 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, 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:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

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

Published:

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

METHODS AND APPARATUSES FOR DECODER-SIDE MOTION VECTOR REFINEMENT IN VIDEO CODING

CROSS-REFERENCE TO RELATED APPLICATION

[0001] The present application claims priority to U.S. Provisional Application No. 62/879,459, entitled "Decoder-side Motion Vector Refinement for Video Coding," filed on July 27, 2019, which is incorporated by reference in its entirety for all purposes.

FIELD

[0002] The present application generally relates to video coding and compression, and in particular but not limited to, methods and apparatuses for decoder-side motion vector refinement (DMVR) in video coding.

BACKGROUND

[0003] Digital video is supported by a variety of electronic devices, such as digital televisions, laptop or desktop computers, tablet computers, digital cameras, digital recording devices, digital media players, video gaming consoles, smart phones, video teleconferencing devices, video streaming devices, etc. The electronic devices transmit, receive, encode, decode, and/or store digital video data by implementing video compression/decompression. Digital video devices implement video coding techniques, such as those described in the standards defined by Versatile Video Coding (VVC), Joint Exploration Test Model (JEM), MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Part 10, Advanced Video Coding (AVC), ITU-T H.265/High Efficiency Video Coding (HEVC), and extensions of such standards.

[0004] Video coding generally utilizes prediction methods (e.g., inter-prediction, intra-prediction) that take advantage of redundancy present in video images or sequences. An important goal of video coding techniques is to compress video data into a form that uses a lower bit rate, while avoiding or minimizing degradations to video quality. With ever-evolving

video services becoming available, encoding techniques with better coding efficiency are needed.

[0005] Video compression typically includes performing spatial (intra frame) prediction and/or temporal (inter frame) prediction to reduce or remove redundancy inherent in the video data. For block-based video coding, a video frame is partitioned into one or more slices, each slice having multiple video blocks, which may also be referred to as coding tree units (CTUs). A CTU may be split into coding units (CUs) using a quadtree with a nested multi-type tree structure, with a CU defining a region of pixels sharing the same prediction mode. Each CTU may contain one coding unit (CU) or recursively split into smaller CUs until the predefined minimum CU size is reached. Each CU (also named leaf CU) contains one or multiple transform units (TUs) and each CU also contains one or multiple prediction units (PUs). Each CU can be coded in intra, inter or IBC modes. Video blocks in an intra coded (I) slice of a video frame are encoded using spatial prediction with respect to reference samples in neighbor blocks within the same video frame. Video blocks in an inter coded (P or B) slice of a video frame may use spatial prediction with respect to reference samples in neighbor blocks within the same video frame or temporal prediction with respect to reference samples in other previous and/or future reference video frames.

[0006] In some examples of the present disclosure, the term 'unit' defines a region of an image covering all components such as luma and chroma; the term 'block' is used to define a region covering a particular component (e.g. luma), and the blocks of different components (e.g. luma vs. chroma) may differ in spatial location when considering the chroma sampling format such as 4:2:0.

[0007] Spatial or temporal prediction based on a reference block that has been previously encoded, e.g., a neighbor block, results in a predictive block for a current video block to be coded. The process of finding the reference block may be accomplished by block matching algorithm. Residual data representing pixel differences between the current block to be coded and the predictive block is referred to as a residual block or prediction errors. An inter-coded block is encoded according to a motion vector that points to a reference block in a reference

frame forming the predictive block, and the residual block. The process of determining the motion vector is typically referred to as motion estimation. An intra coded block is encoded according to an intra prediction mode and the residual block. For further compression, the residual block is transformed from the pixel domain to a transform domain, e.g., frequency domain, resulting in residual transform coefficients, which may then be quantized. The quantized transform coefficients, initially arranged in a two-dimensional array, may be scanned to produce a one-dimensional vector of transform coefficients, and then entropy encoded into a video bitstream to achieve even more compression.

[0008] The encoded video bitstream is then saved in a computer-readable storage medium (e.g., flash memory) to be accessed by another electronic device with digital video capability or directly transmitted to the electronic device wired or wirelessly. The electronic device then performs video decompression (which is an opposite process to the video compression described above) by, e.g., parsing the encoded video bitstream to obtain syntax elements from the bitstream and reconstructing the digital video data to its original format from the encoded video bitstream based at least in part on the syntax elements obtained from the bitstream, and renders the reconstructed digital video data on a display of the electronic device.

[0009] With digital video quality going from high definition, to 4Kx2K or even 8Kx4K, the amount of vide data to be encoded/decoded grows exponentially. It is a constant challenge in terms of how the video data can be encoded/decoded more efficiently while maintaining the image quality of the decoded video data.

SUMMARY

[0010] In general, this disclosure describes examples of techniques relating to decoder-side motion vector refinement (DMVR) in video coding.

[0011] According to a first aspect of the present disclosure, there is provided a method for video coding, including: partitioning a video picture into a plurality of coding units (CUs), wherein each of the plurality of CUs comprises a plurality of sub-blocks; deriving an initial motion vector (MV) for each of the plurality of sub-blocks; deriving a plurality of motion

vector (MV) candidates for bi-directional optical flow (BDOF) and decoder-side motion vector refinement (DMVR); determining an original cost value for the initial MV; determining an updated cost value for the initial MV; obtaining a first threshold and a second threshold for each of the plurality of sub-blocks; determining whether to terminate bi-directional optical flow (BDOF) for each of the plurality of sub-blocks based on the first threshold; and determining whether to terminate decoder-side motion vector refinement (DMVR) for each of the plurality of sub-blocks based on the second threshold.

[0012] According to a second aspect of the present disclosure, there is provided an apparatus for video coding, including: one or more processors; and a memory configured to store instructions executable by the one or more processors; wherein the one or more processors, upon execution of the instructions, are configured to: partition a video picture into a plurality of coding units (CUs), wherein each of the plurality of CUs comprises a plurality of sub-blocks; derive an initial motion vector (MV) for each of the plurality of sub-blocks; derive a plurality of motion vector (MV) candidates for bi-directional optical flow (BDOF) and decoder-side motion vector refinement (DMVR); determine an original cost value for the initial MV; determine an updated cost value for the initial MV; obtain a first threshold and a second threshold for each of the plurality of sub-blocks; determine whether to terminate bi-directional optical flow (BDOF) for each of the plurality of sub-blocks based on the first threshold; and determine whether to terminate decoder-side motion vector refinement (DMVR) for each of the plurality of sub-blocks based on the second threshold.

[0013] According to a third aspect of the present disclosure, there is provided a non-transitory computer readable storage medium, including instructions stored therein, where, upon execution of the instructions by one or more processors, the instructions cause the one or more processors to perform acts including: partitioning a video picture into a plurality of coding units (CUs), wherein each of the plurality of CUs comprises a plurality of sub-blocks; deriving an initial motion vector (MV) for each of the plurality of sub-blocks; deriving a plurality of motion vector (MV) candidates for bi-directional optical flow (BDOF) and decoder-side motion vector refinement (DMVR); determining an original cost value for the initial MV; determining an

updated cost value for the initial MV; obtaining a first threshold and a second threshold for each of the plurality of sub-blocks; determining whether to terminate bi-directional optical flow (BDOF) for each of the plurality of sub-blocks based on the first threshold; and determining whether to terminate decoder-side motion vector refinement (DMVR) for each of the plurality of sub-blocks based on the second threshold.

BRIEF DESCRIPTION OF THE DRAWINGS

[0014] A more particular description of the examples of the present disclosure will be rendered by reference to specific examples illustrated in the appended drawings. Given that these drawings depict only some examples and are not therefore considered to be limiting in scope, the examples will be described and explained with additional specificity and details through the use of the accompanying drawings.

[0015] FIG. 1 is a block diagram illustrating an exemplary video encoder in accordance with some implementations of the present disclosure.

[0016] FIG. 2 is a block diagram illustrating an exemplary video decoder in accordance with some implementations of the present disclosure.

[0017] FIG. 3 is a schematic diagram illustrating an example of decoder-side motion vector refinement (DMVR) in accordance with some implementations of the present disclosure.

[0018] FIG. 4 is a schematic diagram illustrating an example of a DMVR searching procedure in accordance with some implementations of the present disclosure.

[0019] FIG. 5 is a schematic diagram illustrating an example of DMVR integer luma sample searching pattern in accordance with some implementations of the present disclosure.

[0020] FIG. 6 is a block diagram illustrating an exemplary apparatus for video coding in accordance with some implementations of the present disclosure.

[0021] FIG. 7 is a flowchart illustrating an exemplary process of decoder-side motion vector refinement (DMVR) in video coding in accordance with some implementations of the present disclosure.

DETAILED DESCRIPTION

[0022] Reference will now be made in detail to specific implementations, examples of which are illustrated in the accompanying drawings. In the following detailed description, numerous non-limiting specific details are set forth in order to assist in understanding the subject matter presented herein. But it will be apparent to one of ordinary skill in the art that various alternatives may be used. For example, it will be apparent to one of ordinary skill in the art that the subject matter presented herein can be implemented on many types of electronic devices with digital video capabilities.

[0023] The terminology used in the present disclosure is for the purpose of describing exemplary examples only and is not intended to limit the present disclosure. As used in the present disclosure and the appended claims, the singular forms "a," "an" and "the" are intended to include the plural forms as well, unless the context clearly indicates otherwise. It shall also be understood that the terms "or" and "and/or" used herein are intended to signify and include any or all possible combinations of one or more of the associated listed items, unless the context clearly indicates otherwise.

[0024] Reference throughout this specification to "one embodiment," "an embodiment," "an example," "some embodiments," "some examples," or similar language means that a particular feature, structure, or characteristic described is included in at least one embodiment or example. Features, structures, elements, or characteristics described in connection with one or some embodiments are also applicable to other embodiments, unless expressly specified otherwise.

[0025] Throughout the disclosure, the terms "first," "second," "third," and etc. are all used as nomenclature only for references to relevant elements, e.g. devices, components, compositions, steps, and etc., without implying any spatial or chronological orders, unless expressly specified otherwise. For example, a "first device" and a "second device" may refer to two separately formed devices, or two parts, components or operational states of a same device, and may be named arbitrarily.

[0026] As used herein, the term "if" or "when" may be understood to mean "upon" or "in response to" depending on the context. These terms, if appear in a claim, may not indicate that the relevant limitations or features are conditional or optional.

[0027] The terms "module," "sub-module," "circuit," "sub-circuit," "circuitry," "sub-circuitry," "unit," or "sub-unit" may include memory (shared, dedicated, or group) that stores code or instructions that can be executed by one or more processors. A module may include one or more circuits with or without stored code or instructions. The module or circuit may include one or more components that are directly or indirectly connected. These components may or may not be physically attached to, or located adjacent to, one another.

[0028] A unit or module may be implemented purely by software, purely by hardware, or by a combination of hardware and software. In a pure software implementation, for example, the unit or module may include functionally related code blocks or software components, that are directly or indirectly linked together, so as to perform a particular function.

[0029] FIG. 1 shows a block diagram illustrating an exemplary block-based hybrid video encoder 100 which may be used in conjunction with many video coding standards using block-based processing. In the encoder 100, a video frame is partitioned into a plurality of video blocks for processing. For each given video block, a prediction is formed based on either an inter prediction approach or an intra prediction approach. In inter prediction, one or more predictors are formed through motion estimation and motion compensation, based on pixels from previously reconstructed frames. In intra prediction, predictors are formed based on reconstructed pixels in a current frame. Through mode decision, a best predictor may be chosen to predict a current block.

[0030] A prediction residual, representing the difference between a current video block and its predictor, is sent to a Transform circuitry 102. Transform coefficients are then sent from the Transform circuitry 102 to a Quantization circuitry 104 for entropy reduction. Quantized coefficients are then fed to an Entropy Coding circuitry 106 to generate a compressed video bitstream. As shown in FIG. 1, prediction-related information 110 from an inter prediction circuitry and/or an Intra Prediction circuitry 112, such as video block partition info, motion

vectors, reference picture index, and intra prediction mode, are also fed through the Entropy Coding circuitry 106 and saved into a compressed video bitstream 114.

[0031] In the encoder 100, decoder-related circuitries are also needed in order to reconstruct pixels for the purpose of prediction. First, a prediction residual is reconstructed through an Inverse Quantization 116 and an Inverse Transform circuitry 118. This reconstructed prediction residual is combined with a Block Predictor 120 to generate un-filtered reconstructed pixels for a current video block.

[0032] Spatial prediction (or "intra prediction") uses pixels from samples of already coded neighboring blocks (which are called reference samples) in the same video frame as the current video block to predict the current video block.

[0033] Temporal prediction (also referred to as "inter prediction") uses reconstructed pixels from already-coded video pictures to predict the current video block. Temporal prediction reduces temporal redundancy inherent in the video signal. Temporal prediction signal for a given coding unit (CU) or coding block is usually signaled by one or more motion vectors (MVs) which indicate the amount and the direction of motion between the current CU and its temporal reference. Further, if multiple reference pictures are supported, one reference picture index is additionally sent, which is used to identify from which reference picture in the reference picture store the temporal prediction signal comes.

[0034] After spatial and/or temporal prediction is performed, an intra/inter mode decision circuitry 121 in the encoder 100 chooses the best prediction mode, for example based on the rate-distortion optimization method. The block predictor 120 is then subtracted from the current video block; and the resulting prediction residual is de-correlated using the transform circuitry 102 and the quantization circuitry104. The resulting quantized residual coefficients are inverse quantized by the inverse quantization circuitry 116 and inverse transformed by the inverse transform circuitry 118 to form the reconstructed residual, which is then added back to the prediction block to form the reconstructed signal of the CU. Further in-loop filtering 115, such as a deblocking filter, a sample adaptive offset (SAO), and/or an adaptive in-loop filter (ALF) may be applied on the reconstructed CU before it is put in the reference picture store of the

picture buffer 117 and used to code future video blocks. To form the output video bitstream 114, coding mode (inter or intra), prediction mode information, motion information, and quantized residual coefficients are all sent to the entropy coding unit 106 to be further compressed and packed to form the bit-stream.

[0035] For example, a deblocking filter is available in AVC, HEVC as well as the now-current version of VVC. In HEVC, an additional in-loop filter called SAO (sample adaptive offset) is defined to further improve coding efficiency. Another in-loop filter called ALF (adaptive loop filter) is being actively investigated.

[0036] These in-loop filter operations are optional. Performing these operations helps to improve coding efficiency and visual quality. They may also be turned off as a decision rendered by the encoder 100 to save computational complexity.

[0037] It should be noted that intra prediction is usually based on unfiltered reconstructed pixels, while inter prediction is based on filtered reconstructed pixels if these filter options are turned on by the encoder 100.

[0038] FIG. 2 is a block diagram illustrating an exemplary block-based video decoder 200 which may be used in conjunction with many video coding standards. This decoder 200 is similar to the reconstruction-related section residing in the encoder 100 of FIG. 1. In the decoder 200, an incoming video bitstream 201 is first decoded through an Entropy Decoding 202 to derive quantized coefficient levels and prediction-related information. The quantized coefficient levels are then processed through an Inverse Quantization 204 and an Inverse Transform 206 to obtain a reconstructed prediction residual. A block predictor mechanism, implemented in an Intra/inter Mode Selector 212, is configured to perform either an Intra Prediction 208, or a Motion Compensation 210, based on decoded prediction information. A set of unfiltered reconstructed pixels are obtained by summing up the reconstructed prediction residual from the Inverse Transform 206 and a predictive output generated by the block predictor mechanism, using a summer 214.

[0039] The reconstructed block may further go through an In-Loop Filter 209 before it is stored in a Picture Buffer 213 which functions as a reference picture store. The reconstructed video

in the Picture Buffer 213 may be sent to drive a display device, as well as used to predict future video blocks. In situations where the In-Loop Filter 209 is turned on, a filtering operation is performed on these reconstructed pixels to derive a final reconstructed Video Output 222.

[0040] Video coding/decoding standards mentioned above, such as VVC, JEM, HEVC, MPEG-4, Part 10, are conceptually similar. For example, they all use block-based processing. In a Joint Video Experts Team (JVET) meeting, the JVET defined the first draft of the Versatile Video Coding (VVC) and the VVC Test Model 1 (VTM1) encoding method. It was decided to include a quadtree with nested multi-type tree using binary and ternary splits coding block structure as the initial new coding feature of VVC.

Decoder-side Motion Vector Refinement (DMVR) in VVC

[0041] Decoder-side Motion Vector Refinement (DMVR) is a technique for blocks coded in bi-prediction Merge mode. Under this mode, the two motion vectors (MV) of the block can be further refined using bilateral matching (BM) prediction.

[0042] FIG. 3 is a schematic diagram illustrating an example of decoder-side motion vector refinement (DMVR). As shown in the FIG. 3, the bilateral matching method is used to refine motion information of a current CU 322 in the current picture 320 by searching the closest match between its two reference blocks 302, 312 along the motion trajectory of the current CU 322 in its two associated reference pictures, i.e. refPic in List L0 300 and refPic in List L1 310. The patterned rectangular blocks 322, 302, and 312, indicate the current CU and its two reference blocks based on the initial motion information from Merge mode. The patterned rectangular blocks 304, 314 indicate one pair of reference blocks based on a MV candidate used in the motion refinement search process, i.e. motion vector refinement process.

[0043] The MV differences between the MV candidate, i.e. MV0' and MV1', and the initial MV, i.e. MV0 and MV1 (also called the original MV), are MV_{diff} and $-MV_{diff}$, respectively. The MV candidate and the initial are both bi-directional motion vectors. During DMVR, a number of such MV candidates around the initial MV may be checked. Specifically, for each

given MV candidate, its two associated reference blocks may be located in its reference pictures in List 0 and List 1 respectively, and the difference between them may be calculated.

[0044] The block difference may also be referred to as a cost value, and is usually measured in sum of absolute differences (SAD), or row-subsampled SAD (i.e. the SAD calculated with every other row of the block involved). In some other examples, mean-removed SAD, or sum of squared differences (SSD) may also be used as the cost value. The MV candidate with the lowest cost value, or SAD, between its two reference blocks becomes the refined MV and is used to generate the bi-predicted signal as the actual prediction for the current CU.

[0045] In VVC, the DMVR is applied to a CU that satisfies the following conditions:

- The CU is coded with CU level merge mode (not subblock merge mode) with bi-prediction MV;
- With respect to the current picture, one reference picture of the CU is in the past (i.e. with a POC smaller than the current picture POC) and the other reference picture is in the future (i.e. with a POC greater than the current picture POC);
- The POC distances (i.e. absolute POC difference) from both reference pictures to the current picture are the same; and
- The CU has more than 64 luma samples in size and the CU height is more than 8 luma samples.

[0046] The refined MV derived by the DMVR process is used to generate the inter prediction samples and also used in temporal motion vector prediction for future picture coding. While the original MV is used in deblocking process and also in spatial motion vector prediction for future CU coding.

Searching Scheme in DMVR

[0047] As shown in FIG. 3, the MV candidates (or search points) are surrounding the initial MV, with the MV offset obeying the MV difference mirroring rule. In other words, any points that are checked by DMVR, denoted by candidate MV pair (MV0, MV1), obey the following two equations:

$$MV0' = MV0 + MV_{diff}$$

$$MV1' = MV1 - MV_{diff},$$

[0048] where MV_{diff} represents the refinement offset between the initial MV and the refined MV in one of the reference pictures. In current VVC, the refinement search range is two integer luma samples from the initial MV.

[0049] FIG. 4 illustrates an example of the searching process of DMVR. As shown in FIG. 4, the searching process includes the integer sample offset search stage 402 and the fractional sample refinement stage 404.

[0050] To reduce the search complexity, a fast searching method with early termination mechanism is applied in the integer sample offset search stage 402. Instead of a 25 points full search, a 2-iteration search scheme is applied to reduce the number of SAD checking points. FIG. 5 illustrates an example of DMVR integer luma sample searching pattern for the integer sample offset search stage 402. Each rectangular box in FIG. 5 represents a point (MV). As shown in FIG. 5, a maximum of 6 SADs (SADs for Center and P1 ~ P5) are checked in the first iteration according to the fast search method. In the first iteration, the initial MV is Center. First, the SADs of the five points (Center and P1 ~ P4) are compared. If the SAD of Center (i.e. the center position) is smallest, the integer sample offset search stage 402 of DMVR is terminated. Otherwise, one more position P5 (determined based on the SAD distribution of P1 ~ P4), is checked. Then, the position (among P1 ~ P5) with the smallest SAD is selected as the center position of the second iteration search. The process of the second iteration search is the same as that of the first iteration search. The SADs calculated in the first iteration may be re-used in the second iteration, and therefore SADs of only 3 additional points may need to be calculated in the second iteration. It is noted that when the SAD of the center point in the first iteration is smaller than the number of samples used to calculate the SAD (which is equal to w * h / 2, where w and h represent the width and height of the DMVR operation unit, respectively), the whole DMVR process is early terminated without further search.

[0051] The integer sample search 402 is followed by the fractional sample refinement 404. To reduce calculation complexity, the fractional sample refinement 404 is derived using

parametric error surface equation, instead of additional search with SAD comparison. The fractional sample refinement 404 is conditionally invoked based on the output of the integer sample search stage. When the integer sample search stage 402 is terminated with the center having the smallest SAD in either the first iteration or the second iteration search, the fractional sample refinement is further applied.

[0052] In parametric error surface based fractional sample refinement, the SAD costs (or cost values) of the center position and its four neighboring positions are used to fit a 2-D parabolic error surface equation of the following form:

$$E(x,y) = A(x - x_{min})^2 + B(y - y_{min})^2 + C,$$

where (x_{min}, y_{min}) corresponds to the fractional position with the least SAD cost and C corresponds to the minimum cost value. By solving the above equations using the SAD cost values of the five search points, the (x_{min}, y_{min}) may be derived by:

$$x_{min} = (E(-1,0) - E(1,0))/(2(E(-1,0) + E(1,0) - 2E(0,0))) \quad (1)$$

$$y_{min} = (E(0,-1) - E(0,1))/(2((E(0,-1) + E(0,1) - 2E(0,0)) \quad (2)$$

[0053] The values of x_{min} and y_{min} are further constrained to be between – 8 and 8, which corresponds to half-pel offset from the center point with 1/16th-pel MV accuracy. The computed fractional offset (x_{min}, y_{min}) is added to the integer distance MV refinement to get the sub-pixel accuracy MV refinement.

Bilinear-interpolation and sample padding for DMVR

[0054] In VVC, the resolution of the MVs is 1/16 luma samples. The samples at the fractional position are interpolated using an 8-tap interpolation filter. In DMVR search, when a candidate MV points to a sub-pel location, those related fractional position samples need to be interpolated. To reduce the calculation complexity, the bi-linear interpolation filter is used to generate the fractional samples in the searching process in DMVR.

[0055] Another effect from using bi-linear filter for interpolation is that with a 2-sample search range, the DVMR search process does not access more reference samples compared to the normal motion compensation process. After the refined MV is attained by the DMVR search

process, the normal 8-tap interpolation filter is applied to generate the final prediction. Again, in this 8-tap interpolation process, sample padding is used to avoid accessing more reference samples than normal motion compensation process. More specifically, in the 8-tap interpolation process based on a refine MV, samples that are beyond those needed for the motion compensation based on the original MV will be padded from their neighboring available samples.

Maximum DMVR processing unit

[0056] When the width and/or height of a CU are larger than 16 luma samples, the DMVR operation for the CU is performed based on DMVR processing units with maximum width and/or height equal to 16 samples. In other words, in such case, the original CU is partitioned into sub-blocks with width and/or height equal to 16 luma samples for DMVR operation. The maximum processing unit size for the DMVR searching process is limited to 16x16.

[0057] In the current VVC design, there is no control flag to control the enabling of the DMVR. However, it is not guaranteed that the DMVR refined MV is always better than the one before the refinement. In some cases, the DMVR refinement process may produce a refined MV that is worse than the original one. According to some examples of the present disclosure, several methods are proposed to reduce the penalty resulted from such uncertainty of DMVR MV refinement.

The early termination of DMVR and BDOF

[0058] In the current VVC specification, the SAD based early termination method is applied to reduce the computational complexity of the DMVR and BDOF by partially bypassing certain DMVR/BDOF processes based on the SAD value between L0 and L1 prediction samples.

[0059] Firstly, when both DMVR and BDOF are enabled, for each subblock (e.g., a 16x16 subblock), the SAD value that is calculated at the center searching position, i.e., offset (0, 0), is used to early terminate the whole BDOF process of the subblock. Specifically, if the SAD of the subblock is greater than or equal to one predefined threshold, the BDOF is applied to the subblock after the DMVR of the subblock is done; otherwise (i.e., the SAD is less than the

threshold), the BDOF of the subblock will be completely skipped. In one example, the threshold, i.e. *threshold*, to determine whether the BDOF of one subblock is applicable is calculated as:

$$threshold = ((subWidth >> 2) * (subHeight >> 2) * bdofBlkDiffThres) >> 5$$

$$bdofBlkDiffThres = 1 << (BitDepth - 3 + shift), and\ shift = Max(2, 14 - BitDepth)$$

where *BitDepth* is the internal bit-depth used for coding video signal; *subWidth* and *subHeight* are the width and height of the subblock.

[0060] Similarly, during the motion refinement of the DMVR, the same initial SAD value calculated from the center search position, i.e., offset (0, 0), is also used to bypass the following DMVR motion refinement processes. Specifically, if the initial SAD value is greater than or equal to another predefined threshold, the whole DMVR refinement is bypassed. In the current design, the threshold used for the DMVR early termination is set equal to *subWidth* * *subHeight*.

Updated cost values for DMVR by adjusting cost value(s)

[0061] Several exemplary approaches are proposed to favor the original MV during the DMVR process. It is noted that these different approaches may be applied independently or jointly.

[0062] The terms "initial MV" and "original MV" may be used interchangeably in some examples of the present disclosure.

[0063] In some examples, during the DMVR process, the cost values for the initial MV and each of the MV candidates may be adjusted or updated to favor the initial MV. That is, after the calculation of cost values (e.g., SADs) of the searching points in the DMVR process, the cost value(s) may be adjusted to increase the probability of the initial MV having the least cost value among the updated cost values, i.e. to favor the initial MV.

[0064] Thus, after the updated cost values are obtained, the initial MV has a higher chance of being selected as the MV with lowest cost during the DMVR process.

[0065] Here, the SAD values are used as an exemplary cost value for illustrative purpose. Other values, such as row-subsampled SAD, mean-removed SAD, or sum of squared difference (SSD), may also be used as the cost values.

[0066] In some examples, the SAD value between the reference blocks referred by the initial MV (or original MV) is decreased by a first value $Offset_{SAD}$ calculated through a predefined process compared against the SAD values of other MV candidates. Therefore, the initial MV is favored relative to other candidate MVs since its SAD value is decreased.

[0067] In one example, the value of $Offset_{SAD}$ may be determined as 1/N of the SAD value associated with the initial MV, where N is an integer (e.g. 4, 8 or 16).

[0068] In another example, the value of $Offset_{SAD}$ may be determined as a constant value M.

[0069] In yet another example, the value of $Offset_{SAD}$ may be determined according to coded information in the current CU, the coded information including at least one or a combination of: coding block size, magnitude of motion vectors, the SAD of the initial MV, and relative position of a DMVR process unit. For example, the value of $Offset_{SAD}$ may be determined as 1/N of the SAD value associated with the initial MV, where N is an integer value (e.g. 4, 8 or 16) chosen based on the block size of the current CU. When the current block size is larger than or equal to a predefined size (e.g. 16x16), the value of N is set to 8; otherwise, the value of N is set to 4. For example, the value of $Offset_{SAD}$ may be determined as 1/N of the SAD value associated with the initial MV, where N is an integer value (e.g. 4, 8 or 16) chosen based on the distance between the center position of the DMVR process unit and the center position of current CU. When the distance is larger than or equal to a predefined threshold, N is set to one value (e.g. 8); otherwise, N is set to another value (e.g. 4).

[0070] In the examples, decreasing the SAD value associated with the initial MV by a certain value $Offset_{SAD}$ is described. In practice, the idea may be implemented differently. For example, instead of decreasing the SAD value associated with the initial MV, the value of $Offset_{SAD}$ may be added to those SADs associated with other MV candidates during DMVR search process, and the results in these two cases are equivalent.

[0071] In some other examples, the SAD values between the reference blocks referred by the non-initial MV candidates are increased by a second value $Offset_{SAD}$' calculated through a predefined process. The second value $Offset_{SAD}$' and the first value $Offset_{SAD}$ may be the same or different. Therefore, the initial MV is favored since the SAD values of the non-initial MVs are increased.

[0072] In one example, the value of $Offset_{SAD}$' may be determined as 1/N of the SAD value associated with the non-initial MV, where N is an integer (e.g. 4, 8 or 16).

[0073] In another example, the value of $Offset_{SAD}$' may be determined as a constant value M.

[0074] In yet another example, the value of $Offset_{SAD}$' may be determined according to coded information in the current CU, which may include coding block size, the magnitude of the motion vectors, the SAD value of the non-initial MVs, and/or the relative position of DMVR process unit within the current CU. For example, this value may be determined as 1/N of the SAD value from the BM using the non-initial MV, where N is an integer (e.g. 4, 8 or 16) chosen based on block size. When the current block size is larger than or equal to a predefined size (e.g. 16x16), the value of N is set to 8; otherwise, the value of N is set to 4. For example, the value of $Offset_{SAD}$' may be determined as 1/N of the SAD value from the BM using the non-initial MV, where N is an integer value (e.g. 4, 8 or 16) chosen based on the distance between the center position of the DMVR process unit and the center position of current CU. When the distance is larger than or equal to a predefined threshold, N is set to one value (e.g. 8); otherwise, N is set to another value (e.g. 4).

[0075] In the examples, increasing the SAD value associated with the non-initial MV candidates by a certain value $Offset_{SAD}$' is described. In practice, the idea may be implemented differently. For example, instead of increasing the SAD value associated with the non-initial MV, the value of $Offset_{SAD}$' may be subtracted from the SAD associated with the initial MV during the DMVR search process, and the results are equivalent.

[0076] In some further examples, the BM SAD associated with the initial MV is calculated based on a proper subset of the samples used for SAD calculation associated with the non-initial MVs. That is, the SAD value of the initial MV is determined using fewer samples

compared with the SAD values of the MV candidates. This may be similar to decreasing the SAD value of the initial MV.

[0077] According to some examples of the present disclosure, a parameter may be signaled to the decoder for adjusting or updating the cost values for the initial MV and/or each of the MV candidates to favor the initial MV. The value of the parameter may be signaled in the bistream in a sequence parameter set, a picture parameter set, a slice header, a coding tree unit (CTU) and/or a coding unit (CU).

[0078] In some examples, the parameter may be the values used in adjusting the at least one of the cost values described in the above examples, such as N or M. For example, in the cases of deceasing the SAD value of the initial MV, the SAD value of the initial MV may be decreased by the reciprocal of the value of the parameter signaled multiplying the cost value of the initial MV (that is, the value of $Offset_{SAD}$ is determined as 1/N of the SAD value associated with the initial MV), or decreased by the value of the parameter (that is, the value of $Offset_{SAD}$ is determined as a constant value M). A set of codewords may be designed for the signaling of the value N or M. The value of the parameter signaled is selected from a predefined set of values based on the set of codewords, each one of the codewords corresponding to one of the values in the predefined set. In one example, a set of values may be predefined as {4, 8, 16}. The binary codewords may be assigned to each value within the predefined set. An example of binary codewords is illustrated in Table 1 below.

Table 1 An example of codewords indicating parameter values for signaling

Value of Parameter	**Codeword**
4	0
8	10
16	11

[0079] In some other examples, a special value may be signaled into the bitstream in a sequence parameter set, a picture parameter set, a slice header, a CTU and/or a CU, indicating that the initial MV has an updated cost value of zero, which is equivalent to the case of disabling

DMVR. In one example, in the case where the cost value of the initial MV is decreased by $Offset_{SAD}$ and the value of $Offset_{SAD}$ is determined as 1/N of the SAD value associated with the initial MV, where N is an integer, N =1 (i.e., the value of the parameter signaled is 1) will make the SAD associated with the original MV equal to zero. In such case, the refined MV derived by the DMVR process is always the original MV (i.e., the original MV is the refined MV in this case), which is equivalent to disabling the DMVR. In some examples, the special value of one (1) may be included in the predefined set of values of the parameter, which may be, for example, {1, 4, 8, 16}.

[0080] According to the above examples, the DMVR process is modified such that the initial MV is favored compared to the other MV candidates in the integer sample offset search stage, thereby reducing the penalty resulted from the possible scenario that the refined MV is worse than the original MV.

[0081] As discussed earlier, the SAD threshold used to early terminate the BDOF process of one subblock is dependent on the coding bit-depth of the video signal. However, due to the fact that the operational bit-depth of the DMVR is fixed to be 10-bit (i.e., independent of the internal bit-depth), the SAD between the L0 and L1 prediction signals for the DMVR motion refinement is always in the precision of 10-bit. In other words, the value of the existing SAD threshold value may not be perfectly matched with the prediction signal precision to make reliable early termination decision of the DMVR and BDOF. Therefore, to solve such problem, it is proposed to modify the SAD threshold of the BDOF early termination to be independent from the internal bit-depth.

[0082] Specifically, by the proposed method, the threshold used for the BDOF early termination is set equal to $N*subWidth * subHeight$, where subWidth and subHeight are the width and the height of one subblock and N is one constant number which represents per-sample difference between L0 and L1 prediction signals. In practice, different values of the constant number N may be applied which can provide varying tradeoff between the coding efficiency of the BDOF and the encoding/decoding complexity. In one specific example, it is proposed to set the value of N equal to 2, i.e., $threshold = 2*subWidth*subHeight$. In another

example, it is proposed to set the value of N equal to 4, i.e., *threshold = 4*subWidth*subHeight.* On the other hand, though the initial DMVR SAD value is used to determine whether the DMVR and BDOF processes are skipped, two different thresholds are applied for making the decisions for the two tools. In one example, to achieve one more uniform design, it is proposed to use the same SAD threshold for both the early termination decisions of the DMVR and the BDOF. Specifically, by such method, when the initial SAD of the DMVR is greater than or equal to the threshold, then both the DMVR and BDOF will be enabled for the current subblock; otherwise, i.e., the initial SAD is less than the threshold, then both the DMVR and the BDOF will be bypassed.

[0083] In the current DMVR design, the SAD value associated with the initial MV of one subblock is reduced before being compared with the SAD values of other MV candidates. By doing so, the initial MV has one large priority than other MV candidates to be selected as the final MV of one subblock. In details, the resulting SAD value of the initial MV is calculated as

$$SAD_{init} = SAD_{org} - (SAD_{org}) >> 1)$$

where SAD_{org} is the original SAD derived using the initial MV of the subblock. In the current design, SAD_{init} will be used to determine whether the DMVR and the BDOF process of the current subblock should be skipped or not. In this disclosure, instead of the modified SAD value of the initial MV (i.e., SAD_{init}), it is proposed to use the original SAD value, i.e. SAD_{org}, to early terminate the DMVR and the BDOF processes. In another example, it is proposed to use the original SAD of the initial MV (i.e., SAD_{org}) to early terminate the DMVR of one subblock while the modified SAD (i.e., SAD_{init}) is used to early terminate the BDOF of the subblock. In yet another example, it is proposed to use the modified initial SAD (i.e., SAD_{init}) to early terminate the DMVR of the subblock while the original initial SAD (i.e., SAD_{org}) is used to early the BDOF process of the subblock.

[0084] FIG. 6 is a block diagram illustrating an exemplary apparatus for video coding in accordance with some implementations of the present disclosure. The apparatus 600 may be a terminal, such as a mobile phone, a tablet computer, a digital broadcast terminal, a tablet device, or a personal digital assistant.

[0085] As shown in FIG. 6, the apparatus 600 may include one or more of the following components: a processing component 602, a memory 604, a power supply component 606, a multimedia component 608, an audio component 610, an input/output (I/O) interface 612, a sensor component 614, and a communication component 616.

[0086] The processing component 602 usually controls overall operations of the apparatus 600, such as operations relating to display, a telephone call, data communication, a camera operation and a recording operation. The processing component 602 may include one or more processors 620 for executing instructions to complete all or a part of steps of the above method. Further, the processing component 602 may include one or more modules to facilitate interaction between the processing component 602 and other components. For example, the processing component 602 may include a multimedia module to facilitate the interaction between the multimedia component 608 and the processing component 602.

[0087] The memory 604 is configured to store different types of data to support operations of the apparatus 600. Examples of such data include instructions, contact data, phonebook data, messages, pictures, videos, and so on for any application or method that operates on the apparatus 600. The memory 604 may be implemented by any type of volatile or non-volatile storage devices or a combination thereof, and the memory 604 may be a Static Random Access Memory (SRAM), an Electrically Erasable Programmable Read-Only Memory (EEPROM), an Erasable Programmable Read-Only Memory (EPROM), a Programmable Read-Only Memory (PROM), a Read-Only Memory (ROM), a magnetic memory, a flash memory, a magnetic disk or a compact disk.

[0088] The power supply component 606 supplies power for different components of the apparatus 600. The power supply component 606 may include a power supply management system, one or more power supplies, and other components associated with generating, managing and distributing power for the apparatus 600.

[0089] The multimedia component 608 includes a screen providing an output interface between the apparatus 600 and a user. In some examples, the screen may include a Liquid Crystal Display (LCD) and a Touch Panel (TP). If the screen includes a touch panel, the screen

may be implemented as a touch screen receiving an input signal from a user. The touch panel may include one or more touch sensors for sensing a touch, a slide and a gesture on the touch panel. The touch sensor may not only sense a boundary of a touching or sliding actions, but also detect duration and pressure related to the touching or sliding operation. In some examples, the multimedia component 608 may include a front camera and/or a rear camera. When the apparatus 600 is in an operation mode, such as a shooting mode or a video mode, the front camera and/or the rear camera may receive external multimedia data.

[0090] The audio component 610 is configured to output and/or input an audio signal. For example, the audio component 610 includes a microphone (MIC). When the apparatus 600 is in an operating mode, such as a call mode, a recording mode and a voice recognition mode, the microphone is configured to receive an external audio signal. The received audio signal may be further stored in the memory 604 or sent via the communication component 616. In some examples, the audio component 610 further includes a speaker for outputting an audio signal.

[0091] The I/O interface 612 provides an interface between the processing component 602 and a peripheral interface module. The above peripheral interface module may be a keyboard, a click wheel, a button, or the like. These buttons may include but not limited to, a home button, a volume button, a start button and a lock button.

[0092] The sensor component 614 includes one or more sensors for providing a state assessment in different aspects for the apparatus 600. For example, the sensor component 614 may detect an on/off state of the apparatus 600 and relative locations of components. For example, the components are a display and a keypad of the apparatus 600. The sensor component 614 may also detect a position change of the apparatus 600 or a component of the apparatus 600, presence or absence of a contact of a user on the apparatus 600, an orientation or acceleration/deceleration of the apparatus 600, and a temperature change of apparatus 600. The sensor component 614 may include a proximity sensor configured to detect presence of a nearby object without any physical touch. The sensor component 614 may further include an optical sensor, such as a CMOS or CCD image sensor used in an imaging application. In some

examples, the sensor component 614 may further include an acceleration sensor, a gyroscope sensor, a magnetic sensor, a pressure sensor, or a temperature sensor.

[0093] The communication component 616 is configured to facilitate wired or wireless communication between the apparatus 600 and other devices. The apparatus 600 may access a wireless network based on a communication standard, such as WiFi, 4G, or a combination thereof. In an example, the communication component 616 receives a broadcast signal or broadcast related information from an external broadcast management system via a broadcast channel. In an example, the communication component 616 may further include a Near Field Communication (NFC) module for promoting short-range communication. For example, the NFC module may be implemented based on Radio Frequency Identification (RFID) technology, infrared data association (IrDA) technology, Ultra-Wide Band (UWB) technology, Bluetooth (BT) technology and other technology.

[0094] In an example, the apparatus 600 may be implemented by one or more of Application Specific Integrated Circuits (ASIC), Digital Signal Processors (DSP), Digital Signal Processing Devices (DSPD), Programmable Logic Devices (PLD), Field Programmable Gate Arrays (FPGA), controllers, microcontrollers, microprocessors or other electronic elements to perform the above method.

[0095] A non-transitory computer readable storage medium may be, for example, a Hard Disk Drive (HDD), a Solid-State Drive (SSD), Flash memory, a Hybrid Drive or Solid-State Hybrid Drive (SSHD), a Read-Only Memory (ROM), a Compact Disc Read-Only Memory (CD-ROM), a magnetic tape, a floppy disk and etc.

[0096] FIG. 7 is a flowchart illustrating an exemplary process of decoder-side motion vector refinement in video coding in accordance with some implementations of the present disclosure.

[0097] In step 702, the processor 620 partitions a video picture into a plurality of coding units (CUs), wherein each of the plurality of CUs comprises a plurality of sub-blocks.

[0098] In step 704, the processor 620 derives an initial motion vector (MV) for each of the plurality of sub-blocks.

[0099] In step 706, the processor 620 derives a plurality of motion vector (MV) candidates for bi-directional optical flow (BDOF) and decoder-side motion vector refinement (DMVR).

[0100] In step 708, the processor 620 determines original cost values for the initial MV and each of the MV candidates.

[0101] In step 710, the processor 620 determines updated cost values for the initial MV.

[0102] In step 712, the processor 620 obtains a first threshold of the cost values and a second threshold of the cost values.

[0103] In Step 714, the processor 620 determines whether to terminate bi-directional optical flow (BDOF) for each of the plurality of sub-blocks based on the first threshold.

[0104] In Step 716, the processor 620 determines whether to terminate decoder-side motion vector refinement (DMVR) for each of the plurality of sub-blocks based on the second threshold.

[0105] The parameter may be signaled in one or a combination of: a sequence parameter set, a picture parameter set, a slice header, a coding tree unit (CTU), and/or a coding unit (CU).

[0106] The value of the parameter signaled may be selected from a predefined set of values based on a set of codewords, each one of the codewords corresponding to one of the values in the predefined set.

[0107] In some examples, there is provided an apparatus for video coding. The apparatus includes one or more processors 620; and a memory 604 configured to store instructions executable by the one or more processors; where the one or more processors, upon execution of the instructions, are configured to perform a method as illustrated in FIG. 7.

[0108] In some other examples, there is provided a non-transitory computer readable storage medium 604, having instructions stored therein. When the instructions are executed by one or more processors 620, the instructions cause the processors to perform a method as illustrated in FIG. 7.

[0109] The description of the present disclosure has been presented for purposes of illustration, and is not intended to be exhaustive or limited to the present disclosure. Many modifications, variations, and alternative implementations will be apparent to those of ordinary skill in the art

having the benefit of the teachings presented in the foregoing descriptions and the associated drawings.

[0110] The examples were chosen and described in order to explain the principles of the disclosure, and to enable others skilled in the art to understand the disclosure for various implementations and to best utilize the underlying principles and various implementations with various modifications as are suited to the particular use contemplated. Therefore, it is to be understood that the scope of the disclosure is not to be limited to the specific examples of the implementations disclosed and that modifications and other implementations are intended to be included within the scope of the present disclosure.

WHAT IS CLAIMED IS:

1. A method of video coding, comprising:

partitioning a video picture into a plurality of coding units (CUs), wherein each of the plurality of CUs comprises a plurality of sub-blocks;

deriving an initial motion vector (MV) for each of the plurality of sub-blocks;

deriving a plurality of motion vector (MV) candidates for bi-directional optical flow (BDOF) and decoder-side motion vector refinement (DMVR);

determining an original cost value for the initial MV;

determining an updated cost value for the initial MV;

obtaining a first threshold and a second threshold for each of the plurality of sub-blocks;

determining whether to terminate bi-directional optical flow (BDOF) for each of the plurality of sub-blocks based on the first threshold; and

determining whether to terminate decoder-side motion vector refinement (DMVR) for each of the plurality of sub-blocks based on the second threshold.

2. The method of claim 1, further comprising: obtaining the first threshold based on a height and a width of each of the plurality of sub-blocks.

3. The method of claim 1, further comprising: obtaining the first threshold based on an equation: N x W x H, wherein W is a width of each of the plurality of sub-blocks, H is a height of each of the plurality of sub-blocks and N is any positive number.

4. The method of claim 1, further comprising: obtaining the second threshold based on an equation: M x W x H, wherein W is a width of each of the plurality of sub-blocks, H is a height of each of the plurality of sub-blocks and M is any positive number.

5. The method of claim 1, wherein the first threshold is equal to the second threshold.

6. The method of claim 5, further comprising:

when the original cost value or the updated cost value for the initial MV is greater than or equal to the first threshold and the second threshold, enabling both the DMVR and the BDOF.

7. The method of claim 5, further comprising:

when the original cost value or the updated cost value for the initial MV is smaller than the first threshold and the second threshold, bypassing both the DMVR and the BDOF.

8. The method of claim 1, further comprising:

determining whether to terminate decoder-side motion vector refinement (DMVR) for each of the plurality of sub-blocks based on the first threshold and the original cost value for the initial MV; and

determining whether to terminate bi-directional optical flow (BDOF) for each of the plurality of sub-blocks based on the second threshold and the original cost value for the initial MV.

9. The method of claim 1, further comprising:

determining whether to terminate decoder-side motion vector refinement (DMVR) for each of the plurality of sub-blocks based on the first threshold and the original cost value for the initial MV; and

determining whether to terminate bi-directional optical flow (BDOF) for each of the plurality of sub-blocks based on the second threshold and the updated cost value for the initial MV.

10. The method of claim 1, further comprising:

determining whether to terminate decoder-side motion vector refinement (DMVR) for each of the plurality of sub-blocks based on the first threshold and the updated cost value for the initial MV; and

determining whether to terminate bi-directional optical flow (BDOF) for each of the plurality of sub-blocks based on the second threshold and the original cost value for the initial MV.

11. The method of claim 1, wherein the original cost value and the updated cost value comprise sums of absolute differences (SADs).

12. An apparatus for vide coding, comprising:

one or more processors; and

a memory configured to store instructions executable by the one or more processors;

wherein the one or more processors, upon execution of the instructions, are configured to:

partition a video picture into a plurality of coding units (CUs), wherein each of the plurality of CUs comprises a plurality of sub-blocks;

derive an initial motion vector (MV) for each of the plurality of sub-blocks;

derive a plurality of motion vector (MV) candidates for bi-directional optical flow (BDOF) and decoder-side motion vector refinement (DMVR);

determine an original cost value for the initial MV;

determine an updated cost value for the initial MV;

obtain a first threshold and a second threshold for each of the plurality of sub-blocks;

determine whether to terminate bi-directional optical flow (BDOF) for each of the plurality of sub-blocks based on the first threshold; and

determine whether to terminate decoder-side motion vector refinement (DMVR) for each of the plurality of sub-blocks based on the second threshold.

13. The apparatus of claim 12, wherein the one or more processors are configured to obtain the first threshold based on a height and a width of each of the plurality of sub-blocks.

14. The apparatus of claim 12, wherein the one or more processors are configured to obtain the first threshold based on an equation: N x W x H, wherein W is a width of each of the plurality of sub-blocks, H is a height of each of the plurality of sub-blocks and N is any positive number.

15. The apparatus of claim 12, wherein the one or more processors are configured to obtain the second threshold based on an equation: M x W x H, wherein W is a width of each of the plurality of sub-blocks, H is a height of each of the plurality of sub-blocks and M is any positive number.

16. The apparatus of claim 12, wherein the first threshold is equal to the second threshold.

17. The apparatus of claim 16, wherein the one or more processors are configured to enable both the DMVR and the BDOF when the original cost value or the updated cost value for the initial MV is greater than or equal to the first threshold and the second threshold.

18. The apparatus of claim 16, wherein the one or more processors are configured to bypass both the DMVR and the BDOF when the initial cost value or the updated cost value for the DMVR is smaller than the first threshold and the second threshold.

19. The apparatus of claim 12, wherein the one or more processors are configured to:

determine whether to terminate decoder-side motion vector refinement (DMVR) for each of the plurality of sub-blocks based on the first threshold and the original cost value for the initial MV; and

determine whether to terminate bi-directional optical flow (BDOF) for each of the plurality of sub-blocks based on the second threshold and the original cost value for the initial MV.

20. The apparatus of claim 12, wherein the one or more processors are configured to:

determine whether to terminate decoder-side motion vector refinement (DMVR) for each of the plurality of sub-blocks based on the first threshold and the original cost value for the initial MV; and

determine whether to terminate bi-directional optical flow (BDOF) for each of the plurality of sub-blocks based on the second threshold and the updated cost value for the initial MV.

21. The apparatus of claim 12, wherein the one or more processors are configured to:

determine whether to terminate decoder-side motion vector refinement (DMVR) for each of the plurality of sub-blocks based on the first threshold and the updated cost value for the initial MV; and

determine whether to terminate bi-directional optical flow (BDOF) for each of the plurality of sub-blocks based on the second threshold and the original cost value for the initial MV.

22. The apparatus of claim 12, wherein the original cost value and the updated cost value comprise sums of absolute differences (SADs).

23. A non-transitory computer readable storage medium, comprising instructions stored therein, wherein, upon execution of the instructions by one or more processors, the instructions cause the one or more processors to perform acts comprising:

partitioning a video picture into a plurality of coding units (CUs), wherein each of the plurality of CUs comprises a plurality of sub-blocks;

deriving an initial motion vector (MV) for each of the plurality of sub-blocks;

deriving a plurality of motion vector (MV) candidates for bi-directional optical flow (BDOF) and decoder-side motion vector refinement (DMVR);

determining an original cost value for the initial;

determining an updated cost value for the initial MV;

obtaining a first threshold and a second threshold for each of the plurality of sub-blocks;

determining whether to terminate bi-directional optical flow (BDOF) for each of the plurality of sub-blocks based on the first threshold; and

determining whether to terminate decoder-side motion vector refinement (DMVR) for each of the plurality of sub-blocks based on the second threshold.

24. The non-transitory computer readable storage medium of claim 23, wherein the instructions cause the one or more processors to further perform:

obtaining the first threshold based on a height and a width of each of the plurality of sub-blocks.

25. The non-transitory computer readable storage medium of claim 23, wherein the instructions cause the one or more processors to further perform:

obtaining the first threshold based on an equation: N x W x H, wherein W is a width of each of the plurality of sub-blocks, H is a height of each of the plurality of sub-blocks and N is any positive number.

26. The non-transitory computer readable storage medium of claim 23, wherein the instructions cause the one or more processors to further perform:

obtaining the second threshold based on an equation: M x W x H, wherein W is a width of each of the plurality of sub-blocks, H is a height of each of the plurality of sub-blocks and M is any positive number.

27. The non-transitory computer readable storage medium of claim 23, wherein the first threshold is equal to the second threshold.

28. The non-transitory computer readable storage medium of claim 27, wherein the instructions cause the one or more processors to further perform:

when the original cost value or the updated cost value for the initial MV is greater than or equal to the first threshold and the second threshold, enabling both the DMVR and the BDOF.

29. The non-transitory computer readable storage medium of claim 27, wherein the instructions cause the one or more processors to further perform:

when the original cost value or the updated cost value for the initial MV is smaller than the first threshold and the second threshold, bypassing both the DMVR and the BDOF.

30. The non-transitory computer readable storage medium of claim 23, wherein the instructions cause the one or more processors to further perform:

determining whether to terminate decoder-side motion vector refinement (DMVR) for each of the plurality of sub-blocks based on the first threshold and the original cost value for the initial MV; and

determining whether to terminate bi-directional optical flow (BDOF) for each of the plurality of sub-blocks based on the second threshold and the original cost value for the initial MV.

31. The non-transitory computer readable storage medium of claim 23, wherein the instructions cause the one or more processors to further perform:

determining whether to terminate decoder-side motion vector refinement (DMVR) for each of the plurality of sub-blocks based on the first threshold and the original cost value for the initial MV; and

determining whether to terminate bi-directional optical flow (BDOF) for each of the plurality of sub-blocks based on the second threshold and the updated cost value for the initial MV.

32. The non-transitory computer readable storage medium of claim 23, wherein the instructions cause the one or more processors to further perform:

determining whether to terminate decoder-side motion vector refinement (DMVR) for each of the plurality of sub-blocks based on the first threshold and the updated cost value for the initial MV; and

determining whether to terminate bi-directional optical flow (BDOF) for each of the plurality of sub-blocks based on the second threshold and the original cost value for the initial MV.

33. The non-transitory computer readable storage medium of claim 19, wherein the original cost value and the updated cost value comprise sums of absolute differences (SADs).

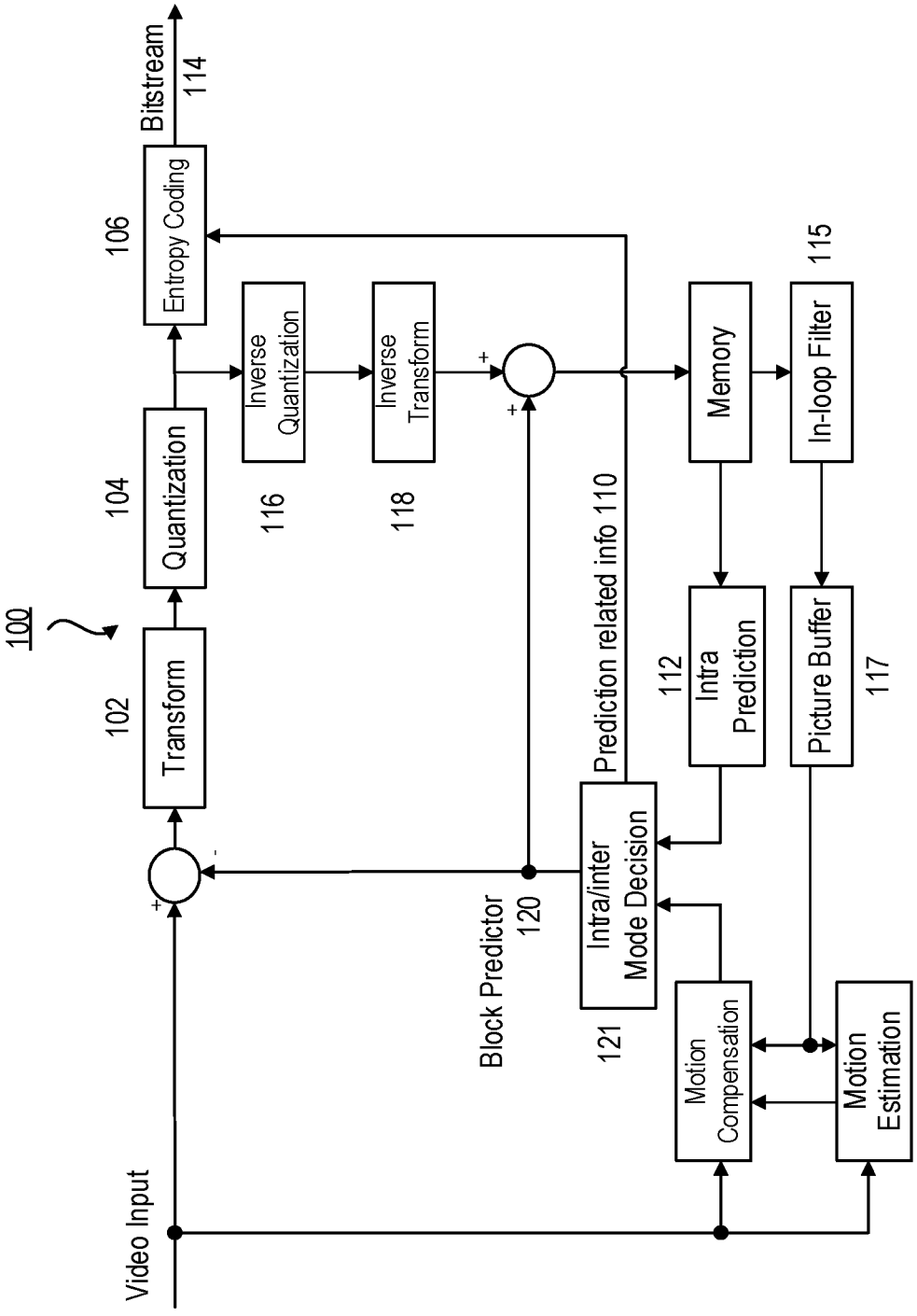
100
Video Input
Transform
102
Quantization
104
Entropy Coding
106
Bitstream
114
Inverse Quantization
116
Inverse Transform
118
Block Predictor
120
Intra/inter Mode Decision
121
Prediction related info 110
Intra Prediction
112
Memory
In-loop Filter
115
Picture Buffer
117
Motion Compensation
Motion Estimation

FIG. 1

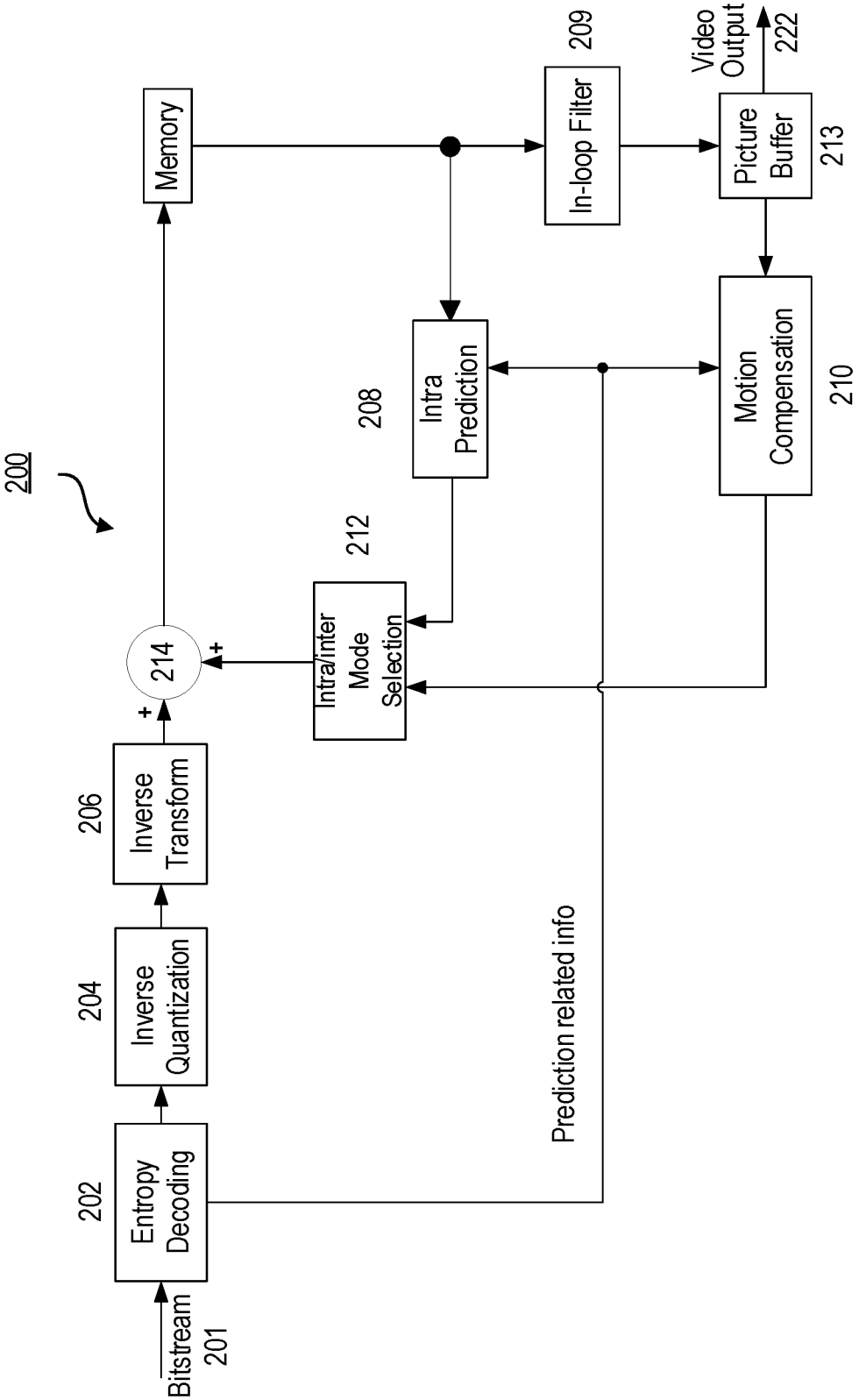
200
Bitstream
201
Entropy Decoding
202
Inverse Quantization
204
Inverse Transform
206
+
214
+
Intra/inter Mode Selection
212
Intra Prediction
208
Memory
In-loop Filter
209
Picture Buffer
213
Video Output
222
Motion Compensation
210
Prediction related info

FIG. 2

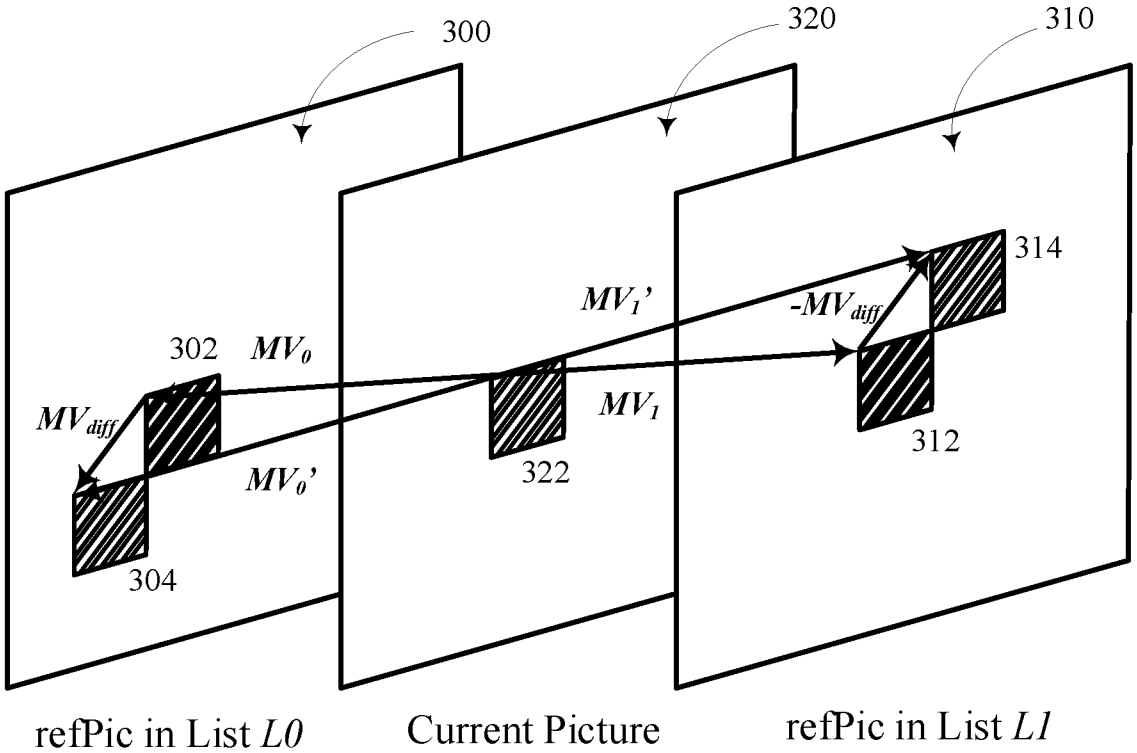
300
320
310
314
MV_1'
$-MV_{diff}$
302
MV_0
MV_{diff}
MV_1
312
MV_0'
322
304
refPic in List *L0*
Current Picture
refPic in List *L1*

FIG. 3

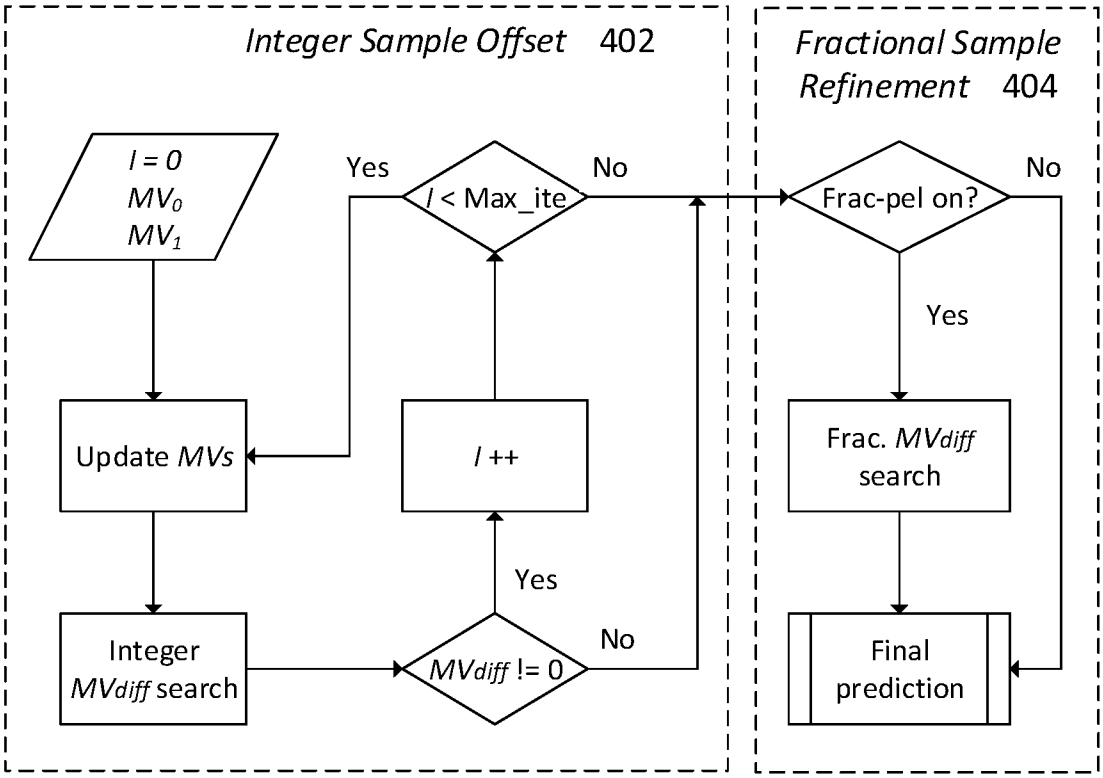
Integer Sample Offset 402
Fractional Sample Refinement 404
I = 0
MV_0
MV_1
Yes
I < Max_ite
No
Frac-pel on?
No
Yes
Update MVs
I ++
Frac. MVdiff search
Yes
Integer MVdiff search
MVdiff != 0
No
Final prediction

FIG. 4

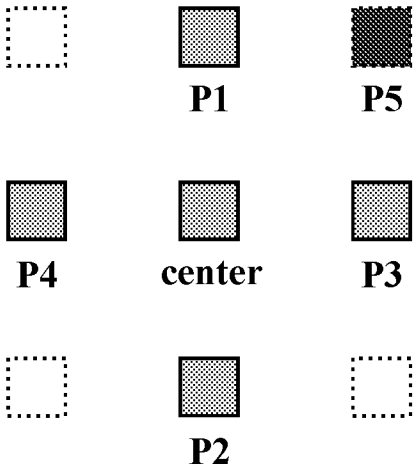
P1
P5
P4
center
P3
P2

FIG. 5

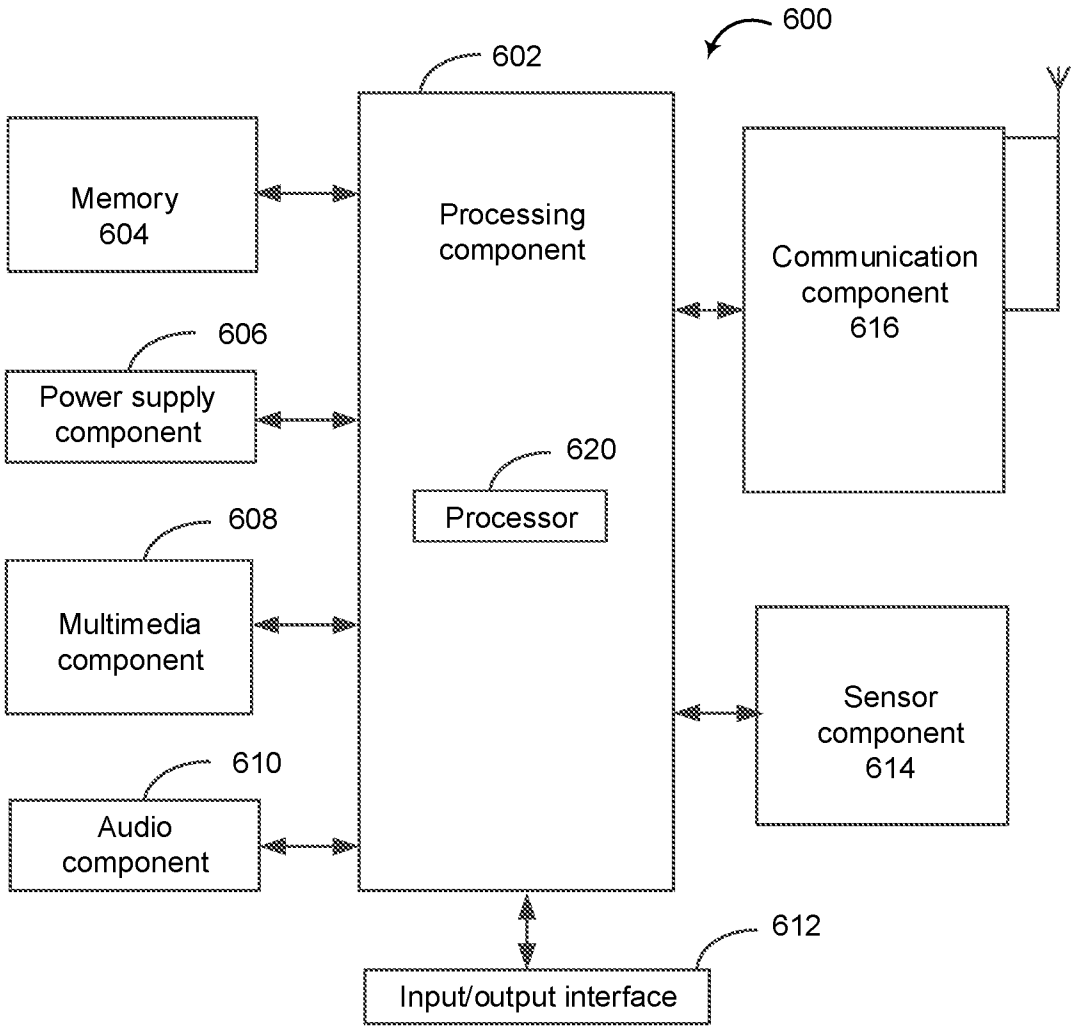
600
602
Memory
604
Processing
component
Communication
component
616
606
Power supply
component
620
Processor
608
Multimedia
component
Sensor
component
614
610
Audio
component
612
Input/output interface

FIG. 6

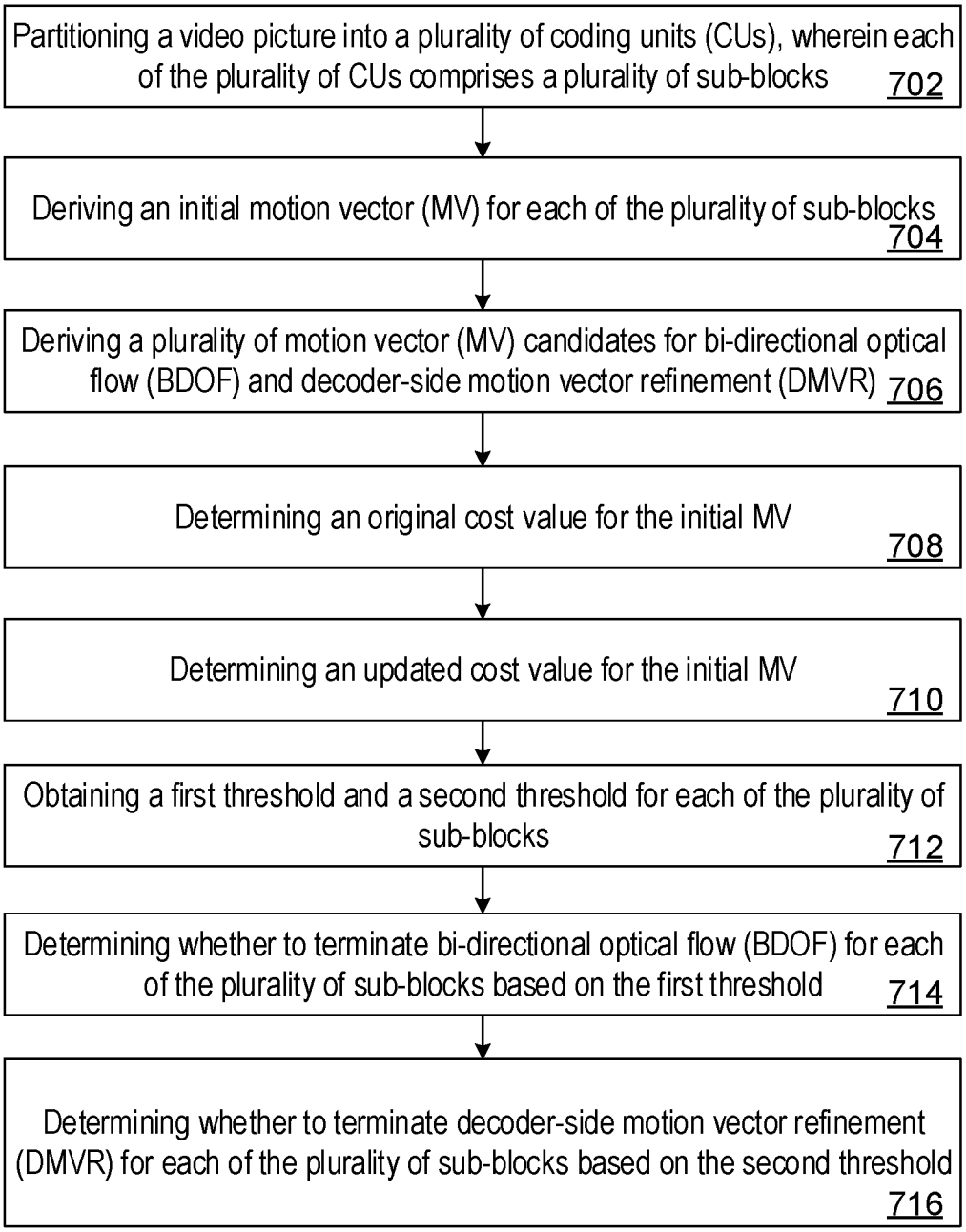
Partitioning a video picture into a plurality of coding units (CUs), wherein each of the plurality of CUs comprises a plurality of sub-blocks 702
Deriving an initial motion vector (MV) for each of the plurality of sub-blocks 704
Deriving a plurality of motion vector (MV) candidates for bi-directional optical flow (BDOF) and decoder-side motion vector refinement (DMVR) 706
Determining an original cost value for the initial MV 708
Determining an updated cost value for the initial MV 710
Obtaining a first threshold and a second threshold for each of the plurality of sub-blocks 712
Determining whether to terminate bi-directional optical flow (BDOF) for each of the plurality of sub-blocks based on the first threshold 714
Determining whether to terminate decoder-side motion vector refinement (DMVR) for each of the plurality of sub-blocks based on the second threshold 716

FIG. 7

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2020/043681

A. CLASSIFICATION OF SUBJECT MATTER

H04N 19/573(2014.01)i, H04N 19/513(2014.01)i, H04N 19/119(2014.01)i, H04N 19/122(2014.01)i, H04N 19/176(2014.01)i, H04N 19/44(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 19/573; H04N 19/513; H04N 19/119; H04N 19/122; H04N 19/176; H04N 19/44

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Korean utility models and applications for utility models
Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & Keywords: DMVR, BDOF, SAD, cost, threshold, decoding

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	KENJI KONDO et al., CE9-related: Early termination of BDOF with DMVR cost, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, [Document: JVET-N0507 (Version 4)], 14th Meeting: Geneva, CH, pp. 1-9, 23 March 2019 [Retrieved on 14-Sep-2020], from <http://phenix.int-evry.fr/jvet> pages 1-2; and figure 2	1,8-11,12,19-22,23,30-33
Y		2-7,13-18,24-29
Y	CHING-YEH CHEN et al., CE9-related: Simplification of cascading DMVR and BDOF processes, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, [Document: JVET-N0097-v1 (Version 2)], 14th Meeting: Geneva, CH, pp. 1-3, 18 March 2019 [Retrieved on 14-Sep-2020], from <http://phenix.int-evry.fr/jvet> page 2; and figure 1	2-7,13-18,24-29
A	BENJAMIN BROSS et al., Versatile Video Coding (Draft 5), Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, [Document: JVET-N1001-v7 (Version 7)], 14th Meeting: Geneva, CH, pp. 1-371, 29 May 2019 [Retrieved on 10-Sep-2020], from <http://phenix.int-evry.fr/jvet> pages 216-217	1-33

☒ 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
"D" document cited by the applicant in the international application
"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
20 October 2020 (20.10.2020)	**21 October 2020 (21.10.2020)**

Name and mailing address of the ISA/KR
International Application Division
Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon, 35208, Republic of Korea
Facsimile No. +82-42-481-8578

Authorized officer
YANG JEONG ROK
Telephone No. +82-42-481-5709

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2020/043681

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JIANLE CHEN et al., Algorithm description for Versatile Video Coding and Test Model 5 (VTM 5), Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, [Document: JVET-N1002-v2 (Version 2)], 14th Meeting: Geneva, CH, pp. 1-371, 11 June 2019 [Retrieved on 10-Sep-2020], from <http://phenix.int-evry.fr/jvet> pages 41-43	1-33
A	WO 2018-130206 A1 (MEDIATEK INC.) 19 July 2018 pages 14-15; claim 1; and figures 5-8	1-33

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2020/043681

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 2018-130206 A1	19/07/2018	CN 110169070 A	23/08/2019
		EP 3566446 A1	13/11/2019
		TW 201832557 A	01/09/2018
		TW I670970 B	01/09/2019
		US 2018-0199057 A1	12/07/2018