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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
01 July 2021 (01.07.2021)

WIPO | PCT

(10) International Publication Number
WO 2021/134017 A1

(51) International Patent Classification:
H04N 19/132 (2014.01) *H04N 19/105* (2014.01)
H04N 19/80 (2014.01) *H04N 19/186* (2014.01)
H04N 19/109 (2014.01) *H04N 19/44* (2014.01)

(21) International Application Number:
PCT/US2020/067017

(22) International Filing Date:
24 December 2020 (24.12.2020)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/953,471 24 December 2019 (24.12.2019) 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)*: **XIU, Xiaoyu** [CN/US]; 6691 Aliso Valley Way, San Diego, California 92130 (US).

(72) Inventors: **CHEN, Yi-Wen**; 8910 University Center Lane, Suite 400, San Diego, California 92122 (US). **MA, Tsung-Chuan**; 3535 Lebon Dr., APT 4315, San Diego, California 92122 (US). **JHU, Hong-Jheng**; Rm101D1-7, 1st Flr, Bldg 1, No. 6, Shangdi West Rd, Haidian District, Beijing, Beijing 100092 (CN). **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, 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, 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,

(54) Title: METHODS AND APPARATUSES FOR REFERENCE PICTURE RESAMPLING

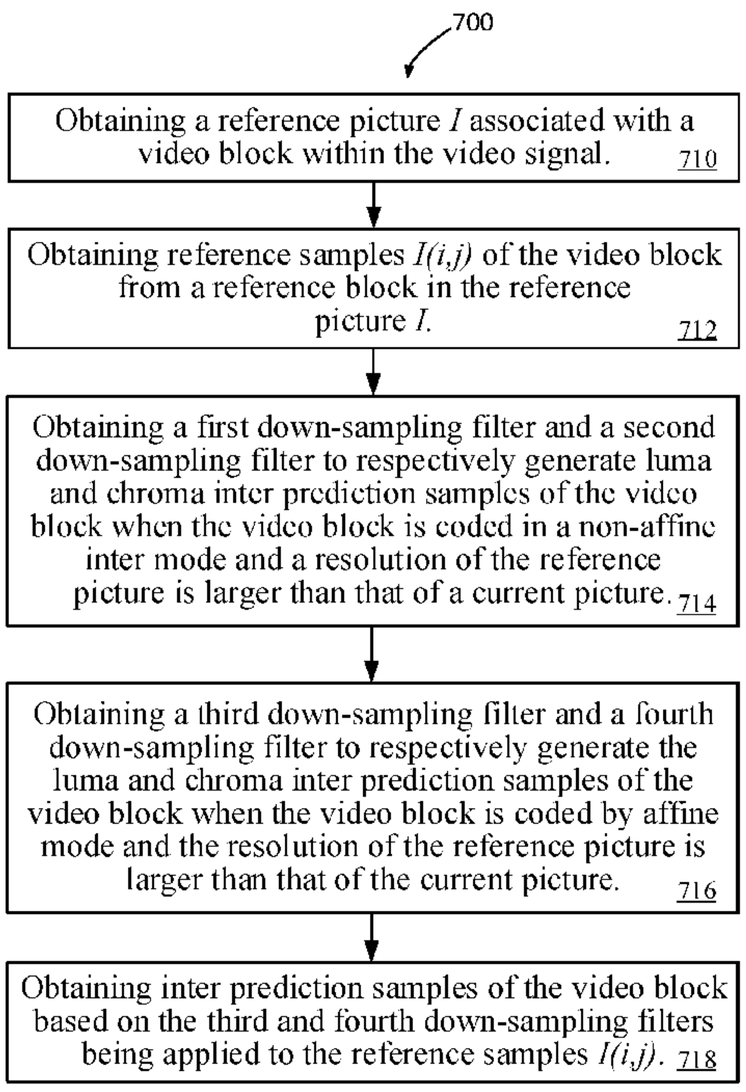


FIG. 7

(57) Abstract: Methods, apparatuses, and non-transitory computer-readable storage mediums are provided for decoding a video signal. A decoder obtains a reference picture *I* associated with a video block within the video signal. The decoder may further obtain reference samples *I (i, j)* of the video block from a reference block in the reference picture *I*. The decoder may also obtain a first down-sampling filter and a second down-sampling filter to respectively generate luma and chroma inter prediction samples of the video block. The decoder may further obtain a third down-sampling filter and a fourth down-sampling filter to respectively generate the luma and chroma inter prediction samples of the video block when the video block is coded by affine mode. The decoder may also obtain inter prediction samples of the video block based on the third and fourth down-sampling filters being applied to the reference samples *I (i, j)*.

[Continued on next page]

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))*

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

METHODS AND APPARATUSES FOR REFERENCE PICTURE RESAMPLING

CROSS-REFERENCE TO RELATED APPLICATION

[0001] This application is based upon and claims priority to Provisional Application No. 62/953,471 filed on December 24, 2019, the entire content thereof is incorporated herein by reference in its entirety for all purposes.

TECHNICAL FIELD

[0002] This disclosure is related to video coding and compression. More specifically, this disclosure relates to methods and apparatus on reference picture resampling technology for video coding.

BACKGROUND

[0003] Various video coding techniques may be used to compress video data. Video coding is performed according to one or more video coding standards. For example, video coding standards include versatile video coding (VVC), joint exploration test model (JEM), high-efficiency video coding (H.265/HEVC), advanced video coding (H.264/AVC), moving picture experts group (MPEG) coding, or the like. Video coding generally utilizes prediction methods (e.g., inter-prediction, intra-prediction, or the like) 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.

SUMMARY

[0004] Examples of the present disclosure provide methods and apparatus for reference picture resampling.

[0005] According to a first aspect of the present disclosure, a method for decoding a video signal is provided. The method may include a decoder obtaining a reference picture I associated with a video block within the video signal. The decoder may also obtain reference samples $I(i, j)$ of the video block from a reference block in the reference picture I. The i and j may represent a coordinate of one sample within the video block. The decoder may further obtain a first down-sampling filter and a second down-sampling filter to respectively generate

luma and chroma inter prediction samples of the video block when the video block is coded in a non-affine inter mode and a resolution of the reference picture I is larger than that of a current picture. The decoder may also obtain a third down-sampling filter and a fourth down-sampling filter to respectively generate the luma and chroma inter prediction samples of the video block when the video block is coded by affine mode and the resolution of the reference picture is larger than that of the current picture. The decoder may further obtain inter prediction samples of the video block based on the third and fourth down-sampling filters being applied to the reference samples $I(i,j)$.

[0006] According to a second aspect of the present disclosure, a computing device is provided. The computing device may include one or more processors, a non-transitory computer-readable memory storing instructions executable by the one or more processors. The one or more processors may be configured to obtain a reference picture I associated with a video block within the video signal. The one or more processors may also be configured to obtain reference samples $I(i,j)$ of the video block from a reference block in the reference picture I. The i and j may represent a coordinate of one sample within the video block. The one or more processors may further be configured to obtain a first down-sampling filter and a second down-sampling filter to respectively generate luma and chroma inter prediction samples of the video block when the video block is coded in a non-affine inter mode and a resolution of the reference picture I is larger than that of a current picture. The one or more processors may also be configured to obtain a third down-sampling filter and a fourth down-sampling filter to respectively generate the luma and chroma inter prediction samples of the video block when the video block is coded by affine mode and the resolution of the reference picture is larger than that of the current picture. The one or more processors may further be configured to obtain inter prediction samples of the video block based on the third and fourth down-sampling filters being applied to the reference samples $I(i,j)$.

[0007] According to a third aspect of the present disclosure, a non-transitory computer-readable storage medium having stored therein instructions is provided. When the instructions are executed by one or more processors of the apparatus, the instructions may cause the apparatus to obtain a reference picture I associated with a video block within the video signal. The instructions may also cause the apparatus to obtain reference samples $I(i,j)$ of the video block from a reference block in the reference picture I. The i and j may represent a coordinate of one sample with the video block. The instructions may further cause the apparatus to obtain a first down-sampling filter and a second down-sampling filter to respectively generate luma

and chroma inter prediction samples of the video block when the video block is coded in a non-affine inter mode and a resolution of the reference picture *I* is larger than that of a current picture. The instructions may also cause the apparatus to obtain a third down-sampling filter and a fourth down-sampling filter to respectively generate the luma and chroma inter prediction samples of the video block when the video block is coded by affine mode and the resolution of the reference picture is larger than that of the current picture. The instructions may further cause the apparatus to obtain inter prediction samples of the video block based on the third and fourth down-sampling filters being applied to the reference samples $I(i, j)$.

[0008] It is to be understood that both the foregoing general description and the following detailed description are examples only and are not restrictive of the present disclosure.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate examples consistent with the present disclosure and, together with the description, serve to explain the principles of the disclosure.

[0010] FIG. 1 is a block diagram of an encoder, according to an example of the present disclosure.

[0011] FIG. 2 is a block diagram of a decoder, according to an example of the present disclosure.

[0012] FIG. 3A is a diagram illustrating block partitions in a multi-type tree structure, according to an example of the present disclosure.

[0013] FIG. 3B is a diagram illustrating block partitions in a multi-type tree structure, according to an example of the present disclosure.

[0014] FIG. 3C is a diagram illustrating block partitions in a multi-type tree structure, according to an example of the present disclosure.

[0015] FIG. 3D is a diagram illustrating block partitions in a multi-type tree structure, according to an example of the present disclosure.

[0016] FIG. 3E is a diagram illustrating block partitions in a multi-type tree structure, according to an example of the present disclosure.

[0017] FIG. 4A is a diagram illustration of a 4-parameter affine model, according to an example of the present disclosure.

[0018] FIG. 4B is a diagram illustration of a 4-parameter affine model, according to an example of the present disclosure.

[0019] FIG. 5 is a diagram illustration of a 6-parameter affine model, according to an example of the present disclosure.

[0020] FIG. 6 is a diagram illustration of an adaptive bit-depth switch, according to an example of the present disclosure.

[0021] FIG. 7 is a method for decoding a video signal, according to an example of the present disclosure.

[0022] FIG. 8 is a method for decoding a video signal, according to an example of the present disclosure

[0023] FIG. 9 is a diagram illustrating a computing environment coupled with a user interface, according to an example of the present disclosure.

DETAILED DESCRIPTION

[0024] Reference will now be made in detail to example embodiments, examples of which are illustrated in the accompanying drawings. The following description refers to the accompanying drawings in which the same numbers in different drawings represent the same or similar elements unless otherwise represented. The implementations set forth in the following description of example embodiments do not represent all implementations consistent with the disclosure. Instead, they are merely examples of apparatuses and methods consistent with aspects related to the disclosure as recited in the appended claims.

[0025] The terminology used in the present disclosure is for the purpose of describing particular embodiments 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 term "and/or" used herein is intended to signify and include any or all possible combinations of one or more of the associated listed items.

[0026] It shall be understood that, although the terms "first," "second," "third," etc. may be used herein to describe various information, the information should not be limited by these terms. These terms are only used to distinguish one category of information from another. For example, without departing from the scope of the present disclosure, first information may be termed as second information; and similarly, second information may also be termed as first information. As used herein, the term "if" may be understood to mean "when" or "upon" or "in response to a judgment" depending on the context.

[0027] The first version of the HEVC standard was finalized in October 2013, which offers

approximately 50% bit-rate saving or equivalent perceptual quality compared to the prior generation video coding standard H.264/MPEG AVC. Although the HEVC standard provides significant coding improvements than its predecessor, there is evidence that superior coding efficiency can be achieved with additional coding tools over HEVC. Based on that, both VCEG and MPEG started the exploration work of new coding technologies for future video coding standardization. A Joint Video Exploration Team (JVET) was formed in Oct. 2015 by ITU-T VECG and ISO/IEC MPEG to begin a significant study of advanced technologies that could enable substantial enhancement of coding efficiency. One reference software called joint exploration model (JEM) was maintained by the JVET by integrating several additional coding tools on top of the HEVC test model (HM).

[0028] In Oct. 2017, the joint call for proposals (CfP) on video compression with capability beyond HEVC was issued by ITU-T and ISO/IEC. In Apr. 2018, 23 CfP responses were received and evaluated at the 10-th JVET meeting, which demonstrated compression efficiency gain over the HEVC around 40%. Based on such evaluation results, the JVET launched a new project to develop the new generation video coding standard that is named as Versatile Video Coding (VVC). In the same month, one reference software codebase, called VVC test model (VTM), was established for demonstrating a reference implementation of the VVC standard.

[0029] Like HEVC, the VVC is built upon the block-based hybrid video coding framework.

[0030] FIG. 1 shows a general diagram of a block-based video encoder for the VVC. Specifically, FIG. 1 shows a typical encoder 100. The encoder 100 has video input 110, motion compensation 112, motion estimation 114, intra/inter mode decision 116, block predictor 140, adder 128, transform 130, quantization 132, prediction related info 142, intra prediction 118, picture buffer 120, inverse quantization 134, inverse transform 136, adder 126, memory 124, in-loop filter 122, entropy coding 138, and bitstream 144.

[0031] 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.

[0032] A prediction residual, representing the difference between a current video block, part of video input 110, and its predictor, part of block predictor 140, is sent to a transform 130 from adder 128. Transform coefficients are then sent from the Transform 130 to a Quantization 132 for entropy reduction. Quantized coefficients are then fed to an Entropy Coding 138 to generate a compressed video bitstream. As shown in FIG. 1, prediction related information 142 from an intra/inter mode decision 116, such as video block partition info, motion vectors (MVs), reference picture index, and intra prediction mode, are also fed through the Entropy Coding

138 and saved into a compressed bitstream 144. Compressed bitstream 144 includes a video bitstream.

[0033] 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 134 and an Inverse Transform 136. This reconstructed prediction residual is combined with a Block Predictor 140 to generate un-filtered reconstructed pixels for a current video block.

[0034] 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.

[0035] 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. The temporal prediction signal for a given coding unit (CU) or coding block is usually signaled by one or more 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 storage, the temporal prediction signal comes from.

[0036] Motion estimation 114 intakes video input 110 and a signal from picture buffer 120 and output, to motion compensation 112, a motion estimation signal. Motion compensation 112 intakes video input 110, a signal from picture buffer 120, and motion estimation signal from motion estimation 114 and output to intra/inter mode decision 116, a motion compensation signal.

[0037] After spatial and/or temporal prediction is performed, an intra/inter mode decision 116 in the encoder 100 chooses the best prediction mode, for example, based on the rate-distortion optimization method. The block predictor 140 is then subtracted from the current video block, and the resulting prediction residual is de-correlated using the transform 130 and the quantization 132. The resulting quantized residual coefficients are inverse quantized by the inverse quantization 134 and inverse transformed by the inverse transform 136 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 122, such as a de-blocking 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 storage of the picture buffer 120 and used to code future video blocks. To form the output video bitstream 144, coding mode (inter

or intra), prediction mode information, motion information, and quantized residual coefficients are all sent to the entropy coding unit 138 to be further compressed and packed to form the bitstream.

[0038] FIG. 1 gives the block diagram of a generic block-based hybrid video encoding system. The input video signal is processed block by block (called CUs). In the VVC, a CU can be up to 128x128 pixels. However, different from the HEVC which partitions blocks only based on quad-trees, in the VVC, one coding tree unit (CTU) is split into CUs to adapt to varying local characteristics based on quad/binary/ternary-tree. Additionally, the concept of multiple partition unit type in the HEVC is removed, i.e., the separation of CU, prediction unit (PU) and transform unit (TU) does not exist in the VVC anymore; instead, each CU is always used as the basic unit for both prediction and transform without further partitions. In the multi-type tree structure, one CTU is firstly partitioned by a quad-tree structure. Then, each quad-tree leaf node can be further partitioned by a binary and ternary tree structure.

[0039] As shown in FIG. 3A, 3B, 3C, 3D, and 3E, there are five splitting types, quaternary partitioning, horizontal binary partitioning, vertical binary partitioning, horizontal ternary partitioning, and vertical ternary partitioning.

[0040] FIG. 3A shows a diagram illustrating block quaternary partition in a multi-type tree structure, in accordance with the present disclosure.

[0041] FIG. 3B shows a diagram illustrating block vertical binary partition in a multi-type tree structure, in accordance with the present disclosure.

[0042] FIG. 3C shows a diagram illustrating block horizontal binary partition in a multi-type tree structure, in accordance with the present disclosure.

[0043] FIG. 3D shows a diagram illustrating block vertical ternary partition in a multi-type tree structure, in accordance with the present disclosure.

[0044] FIG. 3E shows a diagram illustrating block horizontal ternary partition in a multi-type tree structure, in accordance with the present disclosure.

[0045] In FIG. 1, spatial prediction and/or temporal prediction may be performed. Spatial prediction (or "intra prediction") uses pixels from the samples of already coded neighboring blocks (which are called reference samples) in the same video picture/slice to predict the current video block. Spatial prediction reduces spatial redundancy inherent in the video signal. Temporal prediction (also referred to as "inter prediction" or "motion compensated prediction") uses reconstructed pixels from the 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 CU is usually signaled by one or more MVs, which indicate the

amount and the direction of motion between the current CU and its temporal reference. Also, 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 storage the temporal prediction signal comes from. After spatial and/or temporal prediction, the mode decision block in the encoder chooses the best prediction mode, for example, based on the rate-distortion optimization method. The prediction block is then subtracted from the current video block; and the prediction residual is de-correlated using transform and quantized. The quantized residual coefficients are inverse quantized and inverse transformed 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, such as deblocking filter, SAO and ALF may be applied on the reconstructed CU before it is put in the reference picture store and used to code future video blocks. To form the output video bitstream, coding mode (inter or intra), prediction mode information, motion information, and quantized residual coefficients are all sent to the entropy coding unit to be further compressed and packed to form the bitstream.

[0046] FIG. 2 shows a general block diagram of a video decoder for the VVC. Specifically, FIG. 2 shows a typical decoder 200 block diagram. Decoder 200 has bitstream 210, entropy decoding 212, inverse quantization 214, inverse transform 216, adder 218, intra/inter mode selection 220, intra prediction 222, memory 230, in-loop filter 228, motion compensation 224, picture buffer 226, prediction related info 234, and video output 232.

[0047] 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 210 is first decoded through an Entropy Decoding 212 to derive quantized coefficient levels and prediction-related information. The quantized coefficient levels are then processed through an Inverse Quantization 214 and an Inverse Transform 216 to obtain a reconstructed prediction residual. A block predictor mechanism, implemented in an Intra/inter Mode Selector 220, is configured to perform either an Intra Prediction 222 or a Motion Compensation 224, based on decoded prediction information. A set of unfiltered reconstructed pixels is obtained by summing up the reconstructed prediction residual from the Inverse Transform 216 and a predictive output generated by the block predictor mechanism, using a summer 218.

[0048] The reconstructed block may further go through an In-Loop Filter 228 before it is stored in a Picture Buffer 226, which functions as a reference picture store. The reconstructed video in the Picture Buffer 226 may be sent to drive a display device, as well as used to predict future video blocks. In situations where the In-Loop Filter 228 is turned on, a filtering operation is performed on these reconstructed pixels to derive a final reconstructed Video Output 232.

[0049] FIG. 2 gives a general block diagram of a block-based video decoder. The video bitstream is a first entropy decoded at an entropy decoding unit. The coding mode and prediction information are sent to either the spatial prediction unit (if intra coded) or the temporal prediction unit (if inter-coded) to form the prediction block. The residual transform coefficients are sent to the inverse quantization unit and inverse transform unit to reconstruct the residual block. The prediction block and the residual block are then added together. The reconstructed block may further go through in-loop filtering before it is stored in the reference picture storage. The reconstructed video in the reference picture storage is then sent out to drive a display device, as well as used to predict future video blocks.

[0050] The focus of the disclosure is to improve and simplify the existing design of reference picture resampling that is supported in the VVC. In the following, the current coding tools in the VVC that are closely related to the proposed technologies of the disclosure are briefly reviewed.

[0051] **Affine mode**

[0052] In HEVC, only the translation motion model is applied for motion compensated prediction. While in the real world, there are many kinds of motion, e.g., zoom in/out, rotation, perspective motions, and other irregular motions. In the VVC, affine motion compensated prediction is applied by signaling one flag for each inter coding block to indicate whether the translation motion or the affine motion model is applied for inter prediction. In the current VVC design, two affine modes, including 4-parameter affine mode and 6-parameter affine mode, are supported for one affine coding block.

[0053] The 4-parameter affine model has the following parameters: two parameters for translation movement in horizontal and vertical directions, respectively, one parameter for zoom motion and one parameter for rotation motion for both directions. A horizontal zoom parameter is equal to a vertical zoom parameter. The horizontal rotation parameter is equal to the vertical rotation parameter. To achieve a more efficient affine parameter signaling, in the VVC, those affine parameters are derived through two MVs (which are also called control point motion vector (CPMV)) located at the top-left corner and top-right corner of a current block.

[0054] As shown in FIGS. 4A and 4B, the affine motion field of the block is described by two control point MVs (V_0, V_1).

[0055] FIG. 4A shows an illustration of a 4-parameter affine model. FIG. 4B shows an illustration of a 4-parameter affine model. Based on the control point motion, the motion field (v_x, v_y) of one affine coded block is described as

$$v_x = \frac{(v_{1x} - v_{0x})}{w} x - \frac{(v_{1y} - v_{0y})}{w} y + v_{0x}$$
$$v_y = \frac{(v_{1y} - v_{0y})}{w} x + \frac{(v_{1x} - v_{0x})}{w} y + v_{0y} \quad (1)$$

[0056] The 6-parameter affine mode has the following parameters: two parameters for translation movement in horizontal and vertical directions, respectively, one parameter for zoom motion and one parameter for rotation motion in the horizontal direction, one parameter for zoom motion and one parameter for rotation motion in the vertical direction. The 6-parameter affine motion model is coded with three CPMVs.

[0057] FIG. 5 shows an illustration of a 6-parameter affine model. As shown in FIG. 5, three control points of one 6-parameter affine block are located at the top-left, top-right, and bottom left corner of the block. The motion at the top-left control point is related to translation motion, and the motion at the top-right control point is related to rotation and zoom motion in a horizontal direction, and the motion at the bottom-left control point is related to rotation and zoom motion in a vertical direction. Compared to the 4-parameter affine motion model, the rotation and zoom motion in the horizontal direction of the 6-parameter may not be the same as those motion in the vertical direction. Assuming (V_0, V_1, V_2) are the MVs of the top-left, top-right, and bottom-left corners of the current block in FIG. 5, the motion vector of each sub-block (v_x, v_y) is derived using three MVs at control points as:

$$v_{\mathrm{x}} = v_{0x} + (v_{1x} - v_{0x}) * \frac{x}{w} + (v_{2x} - v_{0x}) * \frac{y}{h}$$

$$v_{\mathrm{y}} = v_{0y} + (v_{1y} - v_{0y}) * \frac{x}{w} + (v_{2y} - v_{0y}) * \frac{y}{h} \quad (2)$$

[0058] In the VVC, the CPMVs of affine coding blocks are stored in a separate buffer. The stored CPMVs are only used for the generation of the affine CPMV predictors of affine merge mode (i.e., inheriting affine CPMVs from that of neighboring affine blocks) and affine explicit mode (i.e., signaling the affine CPMVs based on a prediction-based scheme). The sub-block MVs derived from CPMVs are used for motion compensation, MV prediction of translational MVs, and de-blocking.

[0059] Similar to the motion compensation of regular inter blocks, the MVs of each affine subblock may direct to reference samples at fractional sample positions. In such a case, the

interpolation filtering process is needed to generate fractional pixel position reference samples. In order to control the worst-case memory bandwidth requirement and the worst-case computational complexity of interpolation, a set of 6-tap interpolation filters are used for the motion compensation of affine subblocks. Tables 1 and 2 illustrate the interpolation filters that are used for the motion compensations of regular inter blocks and affine blocks, respectively. As can be seen, the 6-tap interpolation filters used for the affine mode are derived directly from the 8-tap filters used for regular inter blocks by directly adding two outermost filter coefficients on each side of 8-tap filters into one single filter coefficient for 6-tap filters, i.e., the filter coefficients P0 and P5 in Table 2 are equal to the sum of the filter coefficients P0 and P1 and the sum of the filter coefficients P6 and P7 in Table 1, respectively.

Table 1 Luma interpolation filters used for regular inter blocks

Fractional position	interpolation filter coefficients							
	P0	P1	P2	P3	P4	P5	P6	P7
1	0	1	−3	63	4	−2	1	0
2	−1	2	−5	62	8	−3	1	0
3	−1	3	−8	60	13	−4	1	0
4	−1	4	−10	58	17	−5	1	0
5	−1	4	−11	52	26	−8	3	−1
6	−1	3	−9	47	31	−10	4	−1
7	−1	4	−11	45	34	−10	4	−1
8	−1	4	−11	40	40	−11	4	−1
9	−1	4	−10	34	45	−11	4	−1
10	−1	4	−10	31	47	−9	3	−1
11	−1	3	−8	26	52	−11	4	−1
12	0	1	−5	17	58	−10	4	−1
13	0	1	−4	13	60	−8	3	−1
14	0	1	−3	8	62	−5	2	−1
15	0	1	−2	4	63	−3	1	0

Table 2 Luma interpolation filters used for affine blocks

Fractional position	interpolation filter coefficients					
	P0	P1	P2	P3	P4	P5
1	1	−3	63	4	−2	1
2	1	−5	62	8	−3	1
3	2	−8	60	13	−4	1
4	3	−10	58	17	−5	1
5	3	−11	52	26	−8	2
6	2	−9	47	31	−10	3
7	3	−11	45	34	−10	3
8	3	−11	40	40	−11	3
9	3	−10	34	45	−11	3
10	3	−10	31	47	−9	2
11	2	−8	26	52	−11	3
12	1	−5	17	58	−10	3
13	1	−4	13	60	−8	2
14	1	−3	8	62	−5	1
15	1	−2	4	63	−3	1

[0060] Additionally, for the motion compensation of chroma samples, the same 4-tap interpolation filters (as illustrated in Table 3) that are used for regular inter blocks are used for affine blocks.

Table 3 Chroma interpolation filters used for inter blocks (i.e., affine blocks and non-affine blocks)

Fractional sample	interpolation filter coefficients			
	P0	P1	P2	P3
1	−1	63	2	0
2	−2	62	4	0
3	-2	60	7	−1
4	−2	58	10	−2
5	−3	57	12	−2
6	−4	56	14	−2
7	−4	55	15	−2
8	−4	54	16	−2
9	−5	53	18	−2
10	−6	52	20	−2
11	−6	49	24	−3
12	−6	46	28	−4
13	−5	44	29	−4
14	−4	42	30	−4
15	−4	39	33	−4
16	−4	36	36	−4
17	−4	33	39	−4
18	−4	30	42	−4
19	−4	29	44	−5
20	−4	28	46	−6
21	−3	24	49	−6
22	−2	20	52	−6
23	−2	18	53	−5
24	−2	16	54	−4
25	−2	15	55	−4
26	−2	14	56	−4
27	−2	12	57	−3
28	−2	10	58	−2
29	−1	7	60	−2
30	0	4	62	−2
31	0	2	63	−1

[0061] Reference picture resampling

[0062] Unlike the HEVC, the emerging VVC standard supports fast spatial resolution switching within the bitstream of one same content. Such capability is referred to as reference picture resampling (RPR) or adaptive resolution switch (ARC). In real-time video applications, allowing resolution change within one coded video sequence without the requirement of inserting a picture that supports random access, or Intra Random Access Point (IRAP) picture (e.g., an IDR picture, or an CRA picture, etc.) can not only adapt the compressed video data to dynamic communication channel condition but also avoid the burst of bandwidth consumption

due to the relatively large size of IDR or CRA pictures. Specifically, the following typical user cases could benefit from the RPR feature:

[0063] Rate adaption in video telephony and conferencing: For adapting the coded video to the changing network conditions, when the network condition gets worse so that available bandwidth becomes lower, the encoder may adapt to it by encoding smaller resolution pictures. Currently, changing picture resolution can be done only after an IRAP picture; this has several issues. An IRAP picture at reasonable quality will be much larger than an inter-coded picture, and will be correspondingly more complex to decode: this costs time and resource. This is a problem if the resolution change is requested by the decoder for loading reasons. It can also break low-latency buffer conditions, forcing an audio re-sync, and the end-to-end delay of the stream will increase, at least temporarily. This can give a poor user experience.

[0064] Active speaker changes in multi-party video conferencing: For multi-party video conferencing, it is common that the active speaker is shown in a bigger video size than the video for the rest of the conference participants. When the active speaker changes, picture resolution for each participant may also need to be adjusted. The need to have ARC feature becomes more important when such a change in active speaker happens frequently.

[0065] Fast start in streaming: For streaming application, it is common that the application would buffer up to a certain length of the decoded picture before start displaying. Starting the bitstream with a smaller resolution would allow the application to have enough pictures in the buffer to start displaying faster.

[0066] Adaptive stream switching in streaming: The Dynamic Adaptive Streaming over HTTP (DASH) specification includes a feature named @mediaStreamStructureId. This enables switching between different representations at open-GOP random access points with non-decodable leading pictures, e.g., CRA pictures with associated RASL pictures in HEVC. When two different representations of the same video have different bit-rates but the same spatial resolution while they have the same value of @mediaStreamStructureId, switching between the two representations at a CRA picture with associated RASL pictures can be performed, and the RASL pictures associated with the switching-at CRA pictures can be decoded with acceptable quality hence enabling seamless switching. With ARC, the @mediaStreamStructureId feature would also be usable for switching between DASH representations with different spatial resolutions.

[0067] At the 15th JVET meeting, the RPR feature was formally supported by the VVC standard. The main aspects of the existing RPR design in the VVC are summarized as follows:

[0068] **RPR high-level signaling**

[0069] According to the current RPR design, in the sequence parameter set (SPS), two syntax elements pic_width_max_in_luma_samples and pic_height_max_in_luma_samples are signaled to specify the maximum width and height of the coded pictures that refer to the SPS. Then, when the picture resolution is changed, one new picture parameter set (PPS) needs to be set when the related syntax elements pic_width_in_luma_samples and pic_height_in_luma_samples are signaled to specify the different picture resolutions of the pictures referring to the PPS. There is bitstream conformance that the values of pic_width_in_luma_sample and pic_height_in_luma_sample should not exceed that of pic_width_max_in_luma_samples and pic_height_max_in_luma_samples. Table 4 illustrates the RPR-related signaling in the SPS and PPS.

Table 4 The RPR signaling in the SPS and PPS

seq_parameter_set_rbsp( ) {	**Descriptor**
...	
pic_width_max_in_luma_samples	ue(v)
pic_height_max_in_luma_samples	ue(v)
...	
}	

pic_parameter_set_rbsp( ) {	**Descriptor**
...	
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
...	
}	

[0070] **Reference picture resampling process**

[0071] When there is a resolution change within one bitstream, a current picture may have one or more reference pictures in different sizes. According to the current RPR design, when picture resolution changes, all MVs to the current picture are normalized to the sample grid of the current picture instead of that of the reference pictures. This can make picture resolution changes transparent to the MV prediction process.

[0072] When picture resolution changes, in addition to MVs, the samples in one reference block have to be up-sampled/down-sampled during the motion compensation of the current

block. In the VVC, the scaling ratio, i.e., refPicWidthInLumaSample/picWidthInLuma and refPicHeightInLumaSample/picHeightInLumaSample, is limited to the range [1/8, 2].

[0073] In the current RPR design, different interpolation filters are applied to interpolate the reference samples when the current picture and its reference picture are in different resolutions. Specifically, when the resolution of the reference picture is equal to or smaller than that of the current picture, the default 8-tap and 4-tap interpolation filters are used to generate the inter prediction samples of luma and chroma samples, respectively. However, the default motion interpolation filters do not present strong low-pass characteristics. When the resolution of the reference picture is higher than that of the current picture, using the default motion interpolation filters will lead to non-negligible aliasing which becomes more severe when the down-sampling ratio is increased. Correspondingly, to improve the inter prediction efficiency of the RPR, two different sets of down-sampling filters are applied when the reference picture is of higher resolution than the current picture. In details, when the down-sampling ratio is equal to or greater than 1.5:1, the following 8-tap and 4-tap Lanczos filters as shown in Table 5 and Table 6 are used.

Table 5 Luma interpolation filters when down-sampling ratio is equal to or greater than 1.5:1

Fractional sample	interpolation filter coefficients							
	P0	P1	P2	P3	P4	P5	P6	P7
0	−1	−5	17	42	17	−5	−1	0
1	0	−5	15	41	19	−5	−1	0
2	0	−5	13	40	21	−4	−1	0
3	0	−5	11	39	24	−4	−2	1
4	0	−5	9	38	26	−3	−2	1
5	0	−5	7	38	28	−2	−3	1
6	1	−5	5	36	30	−1	−3	1
7	1	−4	3	35	32	0	−4	1
8	1	−4	2	33	33	2	−4	1
9	1	−4	0	32	35	3	−4	1
10	1	−3	−1	30	36	5	−5	1
11	1	−3	−2	28	38	7	−5	0
12	1	−2	−3	26	38	9	−5	0
13	1	−2	−4	24	39	11	−5	0
14	0	−1	−4	21	40	13	−5	0
15	0	−1	−5	19	41	15	−5	0

Table 6 Chroma interpolation filters when the down-sampling ratio is equal to or greater than 1.5:1

Fractional sample	interpolation filter coefficients			
	P0	P1	P2	P3
0	12	40	12	0
1	11	40	13	0
2	10	40	15	−1
3	9	40	16	−1
4	8	40	17	−1
5	8	39	18	−1
6	7	39	19	−1
7	6	38	21	−1
8	5	38	22	−1
9	4	38	23	−1
10	4	37	24	−1
11	3	36	25	0
12	3	35	26	0
13	2	34	28	0
14	2	33	29	0
15	1	33	30	0
16	1	31	31	1
17	0	30	33	1
18	0	29	33	2
19	0	28	34	2
20	0	26	35	3
21	0	25	36	3
22	−1	24	37	4
23	−1	23	38	4
24	−1	22	38	5
25	−1	21	38	6
26	−1	19	39	7
27	−1	18	39	8
28	−1	17	40	8
29	−1	16	40	9
30	−1	15	40	10
31	0	13	40	11

[0074] When the down-sampling ratio is equal to or greater than 2:1, the following 8-tap and 4-tap down-sampling filters that are derived by applying a cosine window function to 12-tap SHM down-sampling filters (as shown in Table 7 and Table 8) are used.

Table 7 Luma interpolation filters when the down-sampling ratio is equal to or greater than 2:1

Fractional sample	interpolation filter coefficients							
	P0	P1	P2	P3	P4	P5	P6	P7
0	−4	2	20	28	20	2	−4	0
1	−4	0	19	29	21	5	−4	−2
2	−4	−1	18	29	22	6	−4	−2
3	−4	−1	16	29	23	7	−4	−2
4	−4	−1	16	28	24	7	−4	−2
5	−4	−1	14	28	25	8	−4	−2
6	−3	−3	14	27	26	9	−3	−3
7	−3	−1	12	28	25	10	−4	−3
8	−3	−3	11	27	27	11	−3	−3
9	−3	−4	10	25	28	12	−1	−3
10	−3	−3	9	26	27	14	−3	−3
11	−2	−4	8	25	28	14	−1	−4
12	−2	−4	7	24	28	16	−1	−4
13	−2	−4	7	23	29	16	−1	−4
14	−2	−4	6	22	29	18	−1	−4
15	−2	−4	5	21	29	19	0	−4

Table 8 Chroma interpolation filters when the down-sampling ratio is equal to or greater than 2:1

Fractional sample	interpolation filter coefficients			
	P0	P1	P2	P3
0	17	30	17	0
1	17	30	18	−1
2	16	30	18	0
3	16	30	18	0
4	15	30	18	1
5	14	30	18	2
6	13	29	19	3
7	13	29	19	3
8	12	29	20	3
9	11	28	21	4
10	10	28	22	4
11	10	27	22	5
12	9	27	23	5
13	9	26	24	5
14	8	26	24	6
15	7	26	25	6
16	7	25	25	7
17	6	25	26	7
18	6	24	26	8
19	5	24	26	9
20	5	23	27	9
21	5	22	27	10
22	4	22	28	10
23	4	21	28	11
24	3	20	29	12
25	3	19	29	13
26	3	19	29	13
27	2	18	30	14
28	1	18	30	15
29	0	18	30	16
30	0	18	30	16
31	−1	18	30	17

[0075] At last, the above down-sampling filters are only applied to generate the luma and chroma prediction samples of non-affine inter blocks. For affine mode, the default 8-tap and 4-tap motion interpolation filters are still applied for down-sampling.

[0076] Issues in the Existing RPR Design

[0077] The goals of this disclosure are to improve the coding efficiency of affine mode when the RPR is applied. Specifically, the following issues in the existing RPR design in the VVC are identified:

[0078] First, as discussed earlier, when the resolution of the reference picture is higher than that of the current picture, additional down-sampling filters are only applied to the motion compensation of non-affine modes. For affine mode, the 6-tap and 4-tap motion interpolation filters are applied. Given that those filters are derived from the default motion interpolation filters, they do not show strong low-pass characteristics. Therefore, compared to non-affine mode, the prediction samples of affine mode will present more serious aliasing artifacts due to the famous Nyquist-Shannon sampling theorem. Thus, to achieve better coding performance, it is desirable to also apply appropriate low-pass filters for the motion compensation of affine mode when down-sampling is needed.

[0079] Second, based on the existing RPR design, the fractional pixel position of the reference sample is determined based on the position of the current sample, the MV, and the resolution scaling ratio between the reference picture and the current picture. Correspondingly, when the down-sampling of the reference block is applied, this results in a higher memory bandwidth consumption and computational complexity to interpolate the reference samples of the current block. Assuming the current block is in the size M (width) × N (height). When the reference picture is in the same size as that of the current picture, the integer samples in the size of (M+7) × (N+7) need to be accessed in the reference picture and 8×(M×(N+7)) + 8×M×N multiplications are required for the motion compensation of the current block. If the down-sampling scaling ratio is *s*, the corresponding memory bandwidth and multiplications are increased to (s×M+7) × (s×N+7) and 8×(M×(s×N+7)) + 8×M×N. Table 9 and Table 10 compare the number of integer samples and the number of multiplications per sample that are used for the motion compensation of various block sizes when the RPR down-sampling scaling ratio is 1.5X and 2X, respectively. In Table 9 and Table 10, the columns under the name "RPR 1X" corresponds to the case when the resolutions of the reference picture and the current picture are the same, i.e., the RPR is not applied. The column "ratio to RPR 1X" depicts the corresponding ratio of memory bandwidth/multiplication with an RPR down-sampling ratio larger than 1 to the corresponding worst-case number (i.e., 16×4 bi-prediction) under regular inter mode without the RPR. As can be seen, compared to the worst-case complexity of regular inter prediction, there are significant increases for memory bandwidth and computational complexity when the reference picture has a higher resolution than the current picture. The peak increase comes from 16×4 bi-prediction, where the memory bandwidth and the numbers of multiplications are 231% and 127% of that of worst-case bi-prediction.

Table 9 Per-sample memory bandwidth consumption when RPR ratio is 1.5X and 2X

Width	Height	RPR 1X	RPR 1.5X		RPR 2X	
		Memory bandwidth	Memory bandwidth	ratio to RPR 1X	Memory bandwidth	ratio to RPR 1X
4	8	5.16	7.72	98%	10.78	136%
8	4	5.16	7.72	98%	10.78	136%
4	16	7.91	12.59	159%	18.28	**231%**
16	4	7.91	12.59	159%	18.28	**231%**
4	32	6.70	11.17	141%	16.64	210%
32	4	6.70	11.17	141%	16.64	210%
4	64	6.10	10.46	132%	15.82	200%
64	4	6.10	10.46	132%	15.82	200%
4	128	5.80	10.11	128%	15.41	195%
128	4	5.80	10.11	128%	15.41	195%
8	8	7.03	11.28	143%	16.53	209%
8	16	5.39	9.20	116%	14.02	177%
16	8	5.39	9.20	116%	14.02	177%
8	32	4.57	8.16	103%	12.76	161%
32	8	4.57	8.16	103%	12.76	161%
8	64	4.16	7.64	97%	12.13	153%
64	8	4.16	7.64	97%	12.13	153%
8	128	3.96	7.38	93%	11.81	149%
128	8	3.96	7.38	93%	11.81	149%
16	16	4.13	7.51	95%	11.88	150%
16	32	3.50	6.66	84%	10.82	137%
32	16	3.50	6.66	84%	10.82	137%
16	64	3.19	6.24	79%	10.28	130%
64	16	3.19	6.24	79%	10.28	130%
16	128	3.03	6.02	76%	10.02	127%
128	16	3.03	6.02	76%	10.02	127%
32	32	2.97	5.91	75%	9.85	125%
32	128	2.57	5.34	68%	9.12	115%
128	32	2.57	5.34	68%	9.12	115%
64	64	2.46	5.18	66%	8.90	113%
64	128	2.34	5.00	63%	8.67	110%
128	64	2.34	5.00	63%	8.67	110%
128	128	2.22	4.83	61%	8.44	107%

Table 10 Per-sample number of multiplications when RPR ratio is 1.5X and 2X

		RPR 1X	RPR 1.5X		RPR 2X	
Width	Height	Mul	Mul	ratio to RPR 1X	Mul	ratio to RPR 1X
4	8	23.00	27.00	45%	31.00	52%
8	4	30.00	34.00	57%	38.00	63%
4	16	39.00	47.00	78%	55.00	92%
16	4	60.00	68.00	113%	76.00	**127%**
4	32	35.50	43.50	73%	51.50	86%
32	4	60.00	68.00	113%	76.00	**127%**
4	64	33.75	41.75	70%	49.75	83%
64	4	60.00	68.00	113%	76.00	**127%**
4	128	32.88	40.88	68%	48.88	81%
128	4	60.00	68.00	113%	76.00	**127%**
8	8	46.00	54.00	90%	62.00	103%
8	16	39.00	47.00	78%	55.00	92%
16	8	46.00	54.00	90%	62.00	103%
8	32	35.50	43.50	73%	51.50	86%
32	8	46.00	54.00	90%	62.00	103%
8	64	33.75	41.75	70%	49.75	83%
64	8	46.00	54.00	90%	62.00	103%
8	128	32.88	40.88	68%	48.88	81%
128	8	46.00	54.00	90%	62.00	103%
16	16	39.00	47.00	78%	55.00	92%
16	32	35.50	43.50	73%	51.50	86%
32	16	39.00	47.00	78%	55.00	92%
16	64	33.75	41.75	70%	49.75	83%
64	16	39.00	47.00	78%	55.00	92%
16	128	32.88	40.88	68%	48.88	81%
128	16	39.00	47.00	78%	55.00	92%
32	32	35.50	43.50	73%	51.50	86%
32	128	32.88	40.88	68%	48.88	81%
128	32	35.50	43.50	73%	51.50	86%
64	64	33.75	41.75	70%	49.75	83%
64	128	32.88	40.88	68%	48.88	81%
128	64	33.75	41.75	70%	49.75	83%
128	128	32.88	40.88	68%	48.88	81%

[0080] Third, in the existing RPR design, the VVC only supports the adaptive switch of the resolutions of pictures within one same bitstream while the bit-depth used for coding the video sequence remains the same. However, according to “requirements for a Future Video

Coding Standard" for issuing the CfP of the VVC standard, it is clearly stated "the standard shall support fast representation switching in the case of adaptive streaming services that offer multiple representations of the same content, each having different properties (e.g., spatial resolution or sample bit depth)." In practical video applications, allowing changing the coding bit-depth within a coded video sequence can offer more flexible performance/complexity tradeoffs to video encoders/decoders, especially to software codec implementations due to single-instruction-multiple-data (SIMD) operations.

[0081] Improvements to RPR Coding

[0082] In this disclosure, solutions are proposed to improve the efficiency and reduce the memory bandwidth and computational complexity of the RPR coding in the VVC. More specifically, the technologies proposed in this disclosure can be summarized as follows:

[0083] First, to improve the RPR coding efficiency of the affine mode, new low-pass interpolation filters are proposed to replace the existing 8-tap luma and 4-tap chroma interpolation filters that are used for the affine when the reference picture is in higher resolution than the current picture, i.e., when down-sampling is needed.

[0084] Second, for the simplification of the RPR, it is proposed to disable the RPR-based inter prediction for certain CU sizes that lead to significant memory bandwidth and computational complexity increases compared to that of the regular inter mode without the RPR being applied.

[0085] Third, one approach is proposed to allow dynamic changes of internal bit-depth used for coding one video sequence.

[0086] Down-sampling filters for affine mode

[0087] As mentioned above, the default 6-tap and 4-tap motion interpolation filters are always applied to affine mode no matter whether the resolutions of the current picture and its reference picture are the same or not. Same as the interpolation filters used in the HEVC, the default motion interpolation filters in the VVC do not present strong low-pass characteristics. When the spatial scaling ratio is close to 1, the default motion interpolation filter can provide an acceptable quality of prediction samples. However, when the resolution down-sampling ratio from the reference picture to the current picture becomes larger, based on the Nyquist-Shannon sampling theorem, aliasing artifacts will become much more severe if the same default motion interpolation filter is used. Especially when the applied MV points to reference samples at integer sample positions, the default motion interpolation applies no filter operations at all. This could result in a significant quality drop of the prediction samples for the affine blocks.

[0088] In order to mitigate the aliasing artifacts resulted from the down-sampling, according to the current disclosure, it is proposed to use different interpolation filters with stronger low-pass characteristics to replace the existing default 6-tap/4-tap interpolation filters for the motion compensation of the affine mode. Additionally, in order to maintain the memory bandwidth and computational complexity the same as the regular motion compensation process, the proposed down-sampling filters are in the same length as the existing interpolation filters used for the affine mode, i.e., 6-tap for luma component and 4-tap for chroma component.

[0089] FIG. 7 shows a method for decoding a video signal. The method may be, for example, applied to a decoder.

[0090] In step 710, the decoder may obtain a reference picture I associated with a video block within the video signal.

[0091] In step 712, the decoder may obtain reference samples $I(i,j)$ of the video block from a reference block in the reference picture I. The i and j, for example, may represent a coordinate of one sample within the video block.

[0092] In step 714, the decoder may obtain a first down-sampling filter and a second down-sampling filter to respectively generate luma and chroma inter prediction samples of the video block when the video block is coded in a non-affine inter mode and a resolution of the reference picture I is larger than that of a current picture.

[0093] In step 716, the decoder may obtain a third down-sampling filter and a fourth down-sampling filter to respectively generate the luma and chroma inter prediction samples of the video block when the video block is coded by affine mode and the resolution of the reference picture is larger than that of the current picture.

[0094] In step 718, the decoder may obtain inter prediction samples of the video block based on the third and fourth down-sampling filters being applied to the reference samples $I(i,j)$.

[0095] ***Affine luma down-sampling filters***

[0096] There may be multiple ways to derive luma down-sampling filters for the affine mode.

[0097] **Method 1:** In one or more embodiments of the disclosure, it is proposed to directly derive the luma down-sampling filters of affine mode from the existing luma down-sampling filters of regular inter modes (i.e., non-affine mode). Specifically, by this method, the new 6-tap luma down-sampling filters are derived from the 8-tap luma down-sampling filters in Table 5 (for scaling ratio 1.5X) and Table 7 (for scaling ratio 2X) by adding two left-most/right-most

filter coefficients of the 8-tap filter into one single filter coefficient, respectively, for the 6-tap filter. Table 11 and Table 12 illustrate the proposed 6-tap luma down-sampling filters when the spatial scaling ratio is 1.5:1 and 2:1, respectively.

Table 11 6-tap luma down-sampling filters, scaling ratio is equal to or greater than 1.5:1

Fractional sample	interpolation filter coefficients					
	P0	P1	P2	P3	P4	P5
0	−4	17	42	17	−5	−1
1	−5	15	41	19	−5	−1
2	−5	13	40	21	−4	−1
3	−5	11	39	24	−4	−1
4	−5	9	38	26	−3	−1
5	−5	7	38	28	−2	−2
6	−4	5	36	30	−1	−2
7	−3	3	35	32	0	−3
8	−3	2	33	33	2	−3
9	−3	0	32	35	3	−3
10	−2	−1	30	36	5	−4
11	−2	−2	28	38	7	−5
12	−1	−3	26	38	9	−5
13	−1	−4	24	39	11	−5
14	−1	−4	21	40	13	−5
15	−1	−5	19	41	15	−5

Table 12 6-tap luma down-sampling filters when scaling ratio is equal to or greater than 2:1

Fractional sample	interpolation filter coefficients					
	P0	P1	P2	P3	P4	P5
0	-2	20	28	20	2	−4
1	-4	19	29	21	5	−6
2	−5	18	29	22	6	−6
3	−5	16	29	23	7	−6
4	−5	16	28	24	7	−6
5	−5	14	28	25	8	−6
6	−6	14	27	26	9	−6
7	−4	12	28	25	10	−7
8	−6	11	27	27	11	−6
9	−7	10	25	28	12	−4
10	−6	9	26	27	14	−6
11	−6	8	25	28	14	−5
12	−6	7	24	28	16	−5
13	−6	7	23	29	16	−5
14	−6	6	22	29	18	−5
15	−6	5	21	29	19	-4

[0098] **Method 2:** In one or more embodiments of the disclosure, it is proposed to directly derive the 6-tap affine down-sampling filters from the SHM filters, which are derived based on cosine windowed sinc function. Specifically, in this method, the affine down-sampling filter is derived based on the following equation:

$$Filt_{aff}(n) = h(n) \cdot w(n), \qquad n = -\frac{L-1}{2}, \ldots, \frac{L-1}{2} \tag{3}$$

where L is filter length and $h(n)$ is the frequency response of idea low-pass, which is calculated as:

$$h(n) = s \cdot f_c \cdot sinc(s \cdot f_c \cdot n), \qquad n = -\infty, \ldots, +\infty \tag{4}$$

f_c is a cut-off frequency and s is scaling ratio. $w(n)$ is a cosine window function, which defined as:

$$w(n) = \cos\left(\frac{\pi \cdot n}{L-1}\right), n = -\frac{L-1}{2}, \ldots, \frac{L-1}{2} \tag{5}$$

[0099] In one example, assuming f is 0.9 and L = 6, Table 13 and Table 14 illustrates the derived 6-tap luma down-sampling when the spatial scaling ratio is 1.5X (i.e., $s = 1.5$) and 2X ($s = 2$).

Table 13 6-tap luma down-sampling filters, scaling ratio is equal to or greater than 1.5:1

Fractional sample	Filter coefficients					
	p0	p1	p2	p3	p4	p5
0	-5	18	38	18	-5	0
1	-5	16	39	20	-4	-2
2	-5	14	39	22	-4	-2
3	-5	13	38	24	-3	-3
4	-5	11	37	26	-2	-3
5	-5	9	36	28	-1	-3
6	-5	8	35	30	-1	-3
7	-4	6	34	31	1	-4
8	-4	3	33	33	3	-4
9	-4	1	31	34	6	-4
10	-3	-1	30	35	8	-5
11	-3	-1	28	36	9	-5
12	-3	-2	26	37	11	-5
13	-3	-3	24	38	13	-5
14	-2	-4	22	39	14	-5
15	-2	-4	20	39	16	-5

Table 14 6-tap luma down-sampling filters when scaling ratio is equal to or greater than 2:1

Fractional sample position p	Filter coefficients					
	p0	p1	p2	p3	p4	p5
0	0	19	26	19	0	0
1	-1	18	27	20	2	-2
2	-1	16	27	21	3	-2
3	-2	15	27	22	5	-3
4	-2	14	26	23	6	-3
5	-2	13	26	24	6	-3
6	-3	12	25	25	8	-3
7	-3	11	25	25	9	-3
8	-3	10	25	25	10	-3
9	-3	9	25	25	11	-3
10	-3	8	25	25	12	-3
11	-3	6	24	26	13	-2
12	-3	6	23	26	14	-2
13	-3	5	22	27	15	-2
14	-2	3	21	27	16	-1
15	-2	2	20	27	18	-1

[00100] Note that in Table 13 and Table 14, the filter coefficients are derived in the precision of the 7-bit sign variable, which is kept the same as the down-sampling filters used in the RPR design.

[00101] FIG. 8 shows a method for decoding a video signal. The method may be, for example, applied to a decoder.

[00102] In step 810, the decoder may obtain a frequency response of an ideal low-pass filter based on a cut-off frequency and scaling ratio.

[00103] In step 812, the decoder may obtain a cosine window function based on a filter length.

[00104] In step 814, the decoder may obtain the third down-sampling filter based on the frequency response and the cosine window function.

[00105] ***Affine chroma down-sampling filters***

[00106] In the following, three methods are proposed to down-sample a chroma reference block when the resolution of the reference picture is higher than that of the current picture.

[00107] **Method 1:** In the first method, it is proposed to reuse the existing 4-tap chroma down-sampling filters for 1.5X (Table 6) and 2X (Table 8) designed for non-affine mode under RPR for the down-sampling of the reference samples for the affine mode.

[00108] **Method 2:** In the second method, it is proposed to reuse the default 4-tap chroma interpolation filters (Table 3) to down-sample the reference samples of the affine mode.

[00109] **Method 3:** In the third method, it is proposed to derive chroma down-sampling filters based on the cosine windowed sinc function, as depicted in (3) to (5). Table 15 and Table 16 depict the derived 4-tap chroma down-sampling filters for the scaling ratios of 1.5X and 2X, respectively, when assuming the cut-off frequency of the cosine window sinc function is 0.9.

Table 15 4-tap chroma down-sampling filters, scaling ratio is equal to or greater than 1.5:1

Fractional sample position p	Filter coefficients			
	p0	p1	p2	p3
0	13	38	13	0
1	13	38	14	-1
2	12	38	15	-1
3	11	37	17	-1
4	10	38	18	-2
5	9	38	19	-2
6	8	38	20	-2
7	7	38	21	-2
8	6	37	22	-1
9	6	36	24	-2
10	5	36	25	-2
11	4	35	26	-1
12	4	34	27	-1
13	3	34	28	-1
14	3	33	29	-1
15	2	32	30	0
16	1	31	31	1
17	0	30	32	2
18	-1	29	33	3
19	-1	28	34	3
20	-1	27	34	4
21	-1	26	35	4
22	-2	25	36	5
23	-2	24	36	6
24	-1	22	37	6
25	-2	21	38	7
26	-2	20	38	8
27	-2	19	38	9
28	-2	18	38	10
29	-1	17	37	11
30	-1	15	38	12
31	-1	14	38	13

Table 16 4-tap chroma down-sampling filters, scaling ratio is equal to or greater than 2:1

Fractional sample position p	Filter coefficients			
	p0	p1	p2	p3
0	17	30	17	0
1	17	30	18	-1
2	16	30	18	0
3	16	30	18	0
4	15	30	18	1
5	14	30	18	2
6	13	29	19	3
7	13	29	19	3
8	12	29	20	3
9	11	28	21	4
10	10	28	22	4
11	10	27	22	5
12	9	27	23	5
13	9	26	24	5
14	8	26	24	6
15	7	26	25	6
16	7	25	25	7
17	6	25	26	7
18	6	24	26	8
19	5	24	26	9
20	5	23	27	9
21	5	22	27	10
22	4	22	28	10
23	4	21	28	11
24	3	20	29	12
25	3	19	29	13
26	3	19	29	13
27	2	18	30	14
28	1	18	30	15
29	0	18	30	16
30	0	18	30	16
31	-1	18	30	17

[00110] Constrained block-size for the RPR mode

[00111] As analyzed in the section "problem statement", when down-sampling happens, the existing RPR brings significant complexity increase (e.g., the number of integer samples accessed for motion compensation and the number of multiplications needed). Specifically, the memory bandwidth and the numbers of multiplications when the reference block needs to be down-sampled are 231% and 127% of that of worst-case bi-prediction.

[00112] In one or more embodiments, it is proposed to disable the bi-prediction (but still

allow uni-prediction) during inter prediction for certain block shapes, e.g., 4×N, N×4 and/or 8×8, when the resolution of the reference picture is higher than that of the current picture. Table 17 and Table 18 show the corresponding per-sample memory bandwidth and the number of multiplications when bi-prediction is disabled for 4×N, N×4 and 8×8 block size during the inter prediction with RPR. As can be seen, with the proposed constraint, the memory bandwidth and the numbers of multiplications are reduced to 130% and 107% of that of worst-case bi-prediction for 1.5X down-sampling, and 116% and 113% for 2X down-sampling.

Table 17 Per-sample memory bandwidth consumption after block size constraint is applied for the RPR, for the down-sampling ratios of 1.5X and 2X

		RPR 1X	RPR 1.5X		RPR 2X	
Width	Height	Memory bandwidth	Memory bandwidth	ratio to RPR 1X	Memory bandwidth	ratio to RPR 1X
4	8	5.16	7.72	98%	10.78	136%
8	4	5.16	7.72	98%	10.78	136%
4	16	7.91	10.25	130%	13.09	166%
16	4	7.91	10.25	130%	13.09	166%
4	32	6.70	8.94	113%	11.67	148%
32	4	6.70	8.94	113%	11.67	148%
4	64	6.10	8.28	105%	10.96	139%
64	4	6.10	8.28	105%	10.96	139%
4	128	5.80	7.95	101%	10.61	134%
128	4	5.80	7.95	101%	10.61	134%
8	8	7.03	9.16	116%	11.78	149%
8	16	5.39	9.20	116%	14.02	177%
16	8	5.39	9.20	116%	14.02	177%
8	32	4.57	8.16	103%	12.76	161%
32	8	4.57	8.16	103%	12.76	161%
8	64	4.16	7.64	97%	12.13	153%
64	8	4.16	7.64	97%	12.13	153%
8	128	3.96	7.38	93%	11.81	149%
128	8	3.96	7.38	93%	11.81	149%
16	16	4.13	7.51	95%	11.88	150%
16	32	3.50	6.66	84%	10.82	137%
32	16	3.50	6.66	84%	10.82	137%
16	64	3.19	6.24	79%	10.28	130%
64	16	3.19	6.24	79%	10.28	130%
16	128	3.03	6.02	76%	10.02	127%
128	16	3.03	6.02	76%	10.02	127%
32	32	2.97	5.91	75%	9.85	125%
32	128	2.57	5.34	68%	9.12	115%

128	32	2.57	5.34	68%	9.12	115%
64	64	2.46	5.18	66%	8.90	113%
64	128	2.34	5.00	63%	8.67	110%
128	64	2.34	5.00	63%	8.67	110%
128	128	2.22	4.83	61%	8.44	107%

Table 18 Per-sample number of multiplications when the block size constraint is applied to the RPR, for the down-sampling ratios of 1.5X and 2X

		RPR 1X	RPR 1.5X		RPR 2X	
Width	Height	Mul	Mul	ratio to RPR 1X	Mul	ratio to RPR 1X
4	8	23.00	27.00	45%	31.00	52%
8	4	30.00	34.00	57%	38.00	63%
4	16	39.00	43.00	72%	47.00	78%
16	4	60.00	64.00	107%	68.00	113%
4	32	35.50	39.50	66%	43.50	73%
32	4	60.00	64.00	107%	68.00	113%
4	64	33.75	37.75	63%	41.75	70%
64	4	60.00	64.00	107%	68.00	113%
4	128	32.88	36.88	61%	40.88	68%
128	4	60.00	64.00	107%	68.00	113%
8	8	46.00	50.00	83%	54.00	90%
8	16	39.00	47.00	78%	55.00	92%
16	8	46.00	54.00	90%	62.00	103%
8	32	35.50	43.50	73%	51.50	86%
32	8	46.00	54.00	90%	62.00	103%
8	64	33.75	41.75	70%	49.75	83%
64	8	46.00	54.00	90%	62.00	103%
8	128	32.88	40.88	68%	48.88	81%
128	8	46.00	54.00	90%	62.00	103%
16	16	39.00	47.00	78%	55.00	92%
16	32	35.50	43.50	73%	51.50	86%
32	16	39.00	47.00	78%	55.00	92%
16	64	33.75	41.75	70%	49.75	83%
64	16	39.00	47.00	78%	55.00	92%
16	128	32.88	40.88	68%	48.88	81%
128	16	39.00	47.00	78%	55.00	92%
32	32	35.50	43.50	73%	51.50	86%
32	128	32.88	40.88	68%	48.88	81%
128	32	35.50	43.50	73%	51.50	86%
64	64	33.75	41.75	70%	49.75	83%
64	128	32.88	40.88	68%	48.88	81%
128	64	33.75	41.75	70%	49.75	83%

128	128	32.88	40.88	68%	48.88	81%

[00113] Although in the above example, the RPR mode bi-prediction is only disabled for 4×N, N×4 and 8×8 block sizes, to skilled persons in state-of-the-art of modern video technologies, the proposed constraint is also applicable to other block sizes and inter coding modes (e.g., uni-/bi-prediction, merge/non-merge mode and so forth).

[00114] **Adaptive bit-depth switch**

[00115] In the existing RPR design, the VVC only supports adaptive switch of the resolutions of pictures within one same bitstream while the bit-depth used for coding the video sequence remains the same. However, as analyzed earlier, allowing switch of coding bit-depth within one same bitstream can offer more flexibilities for practical encoder/decoder devices and provide different tradeoffs between coding performance and computational complexity.

[00116] In this section, one adaptive bit-depth switch (ABS) approach is proposed to allow changing the internal coding bit-depth without the requirement of introducing one IRAP picture such as the instantaneous decoder refresh (IDR) picture, etc.

[00117] FIG. 6 depicts one hypothetical example where the current picture 620 and its reference pictures 610 and 630 are coded in different internal bit-depths. FIG. 6 shows reference picture 610 Ref0 with 8-bit coding, current picture 620 with 10-bit coding, and reference picture 630 Ref1 with 12-bit coding. Specifically, in the following, high-level syntax signaling and modifications to the motion compensation process are proposed to the current VVC framework in order to support the proposed ABS capability.

[00118] ***High-level ABS signaling***

[00119] For the proposed ABS signaling, in the SPS, one new syntax element sps_max_bit_depth_minus8 is proposed to replace the existing bit-depth syntax element bit_depth_minus8, which specifies the maximum internal coding bit-depth used for the coded pictures that refer to the SPS. Then, when the coding bit-depth is changed, one new PPS syntax pps_bit_depth_minus8 is sent to specify the different coding bit-depth of the pictures referring to the PPS.

[00120] There is bitstream conformance that the values of pps_bit_depth_minus8 should not exceed that of sps_max_bit_depth_minus8. Table 19 illustrates the proposed ABS signaling in the SPS and PPS.

Table 19 The proposed ABS signaling in the SPS and PPS

seq_parameter_set_rbsp( ) {	**Descriptor**
...	
sps_max_bit_depth_minus8	ue(v)
...	
~~**bit_depth_minus8**~~	~~ue(v)~~
...	
}	

pic_parameter_set_rbsp( ) {	**Descriptor**
...	
pps_bit_depth_minus8	ue(v)
...	
}	

[00121] ***Prediction sample bit-depth adjustment***

[00122] When there is one coding bit-depth change within one coded video sequence, one current picture may be predicted from another reference picture whose reconstructed samples are represented in different bit-depth precisions. When such a case happens, the prediction samples generated from the motion compensation of the reference picture should be adjusted to the coding bit-depth of the current picture.

[00123] ***Interaction with other coding tools***

[00124] Given that when the ABS is applied to the reference picture and the current picture may be represented in different precisions, some existing coding tools in the VVC that utilize the reference samples to derive certain coding parameters may not be working properly. For example, in the current VVC, bi-directional optical flow (BDOF) and decoder-side motion vector refinement (DMVR) are two decoder-side technologies that use temporal prediction samples to improve the inter coding efficiency. Specifically, the BDOF tool utilizes the L0 and L1 prediction samples to calculate sample-wise refinements to improve the prediction sample quality, while the DMVR relies on the L0 and L1 prediction samples to refine the motion vector precision at subblock level. Based on the above consideration, it is proposed to always bypass the BDOF and the DMVR process for one inter block when either of two prediction signals is coded in different bit-depth from the current picture.

[00125] FIG. 9 shows a computing environment 910 coupled with a user interface 960. The computing environment 910 can be part of data processing server. The computing environment

910 includes processor 920, memory 940, and I/O interface 950.

[00126] The processor 920 typically controls overall operations of the computing environment 910, such as the operations associated with the display, data acquisition, data communications, and image processing. The processor 920 may include one or more processors to execute instructions to perform all or some of the steps in the above-described methods. Moreover, the processor 920 may include one or more modules that facilitate the interaction between the processor 920 and other components. The processor may be a Central Processing Unit (CPU), a microprocessor, a single chip machine, a GPU, or the like.

[00127] The memory 940 is configured to store various types of data to support the operation of the computing environment 910. Memory 940 may include predetermine software 942. Examples of such data comprise instructions for any applications or methods operated on the computing environment 910, video datasets, image data, etc. The memory 940 may be implemented by using any type of volatile or non-volatile memory devices, or a combination thereof, such as 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 or optical disk.

[00128] The I/O interface 950 provides an interface between the processor 920 and peripheral interface modules, such as a keyboard, a click wheel, buttons, and the like. The buttons may include but are not limited to, a home button, a start scan button, and a stop scan button. The I/O interface 950 can be coupled with an encoder and decoder.

[00129] In an embodiment, there is also provided a non-transitory computer-readable storage medium comprising a plurality of programs, such as comprised in the memory 940, executable by the processor 920 in the computing environment 910, for performing the above-described methods. For example, the non-transitory computer-readable storage medium may be a ROM, a RAM, a CD-ROM, a magnetic tape, a floppy disc, an optical data storage device or the like.

[00130] The non-transitory computer-readable storage medium has stored therein a plurality of programs for execution by a computing device having one or more processors, where the plurality of programs when executed by the one or more processors, cause the computing device to perform the above-described method for motion prediction.

[00131] In an embodiment, the computing environment 910 may be implemented with one or more application-specific integrated circuits (ASICs), digital signal processors (DSPs), digital signal processing devices (DSPDs), programmable logic devices (PLDs), field-

programmable gate arrays (FPGAs), graphical processing units (GPUs), controllers, micro-controllers, microprocessors, or other electronic components, for performing the above methods.

[00132] 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.

[00133] 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.

CLAIMS

What is claimed is:

1. A method for decoding a video signal, comprising:

obtaining, at a decoder, a reference picture I associated with a video block within the video signal;

obtaining, at the decoder, reference samples $I(i,j)$ of the video block from a reference block in the reference picture I, wherein i and j represent a coordinate of one sample within the video block;

obtaining, at the decoder, a first down-sampling filter and a second down-sampling filter to respectively generate luma and chroma inter prediction samples of the video block when the video block is coded in a non-affine inter mode and a resolution of the reference picture I is larger than that of a current picture;

obtaining, at the decoder, a third down-sampling filter and a fourth down-sampling filter to respectively generate the luma and chroma inter prediction samples of the video block when the video block is coded by affine mode and the resolution of the reference picture is larger than that of the current picture; and

obtaining, at the decoder, inter prediction samples of the video block based on the third and fourth down-sampling filters being applied to the reference samples $I(i,j)$.

2. The method of claim 1, wherein obtaining, at the decoder, the third down-sampling filter and a fourth down-sampling filter comprises:

deriving, at the decoder, the third down-sampling filter is derived from the first down-sampling filter, wherein the third down-sampling filter is used to generate luma inter prediction samples of affine mode, and wherein the first down-sampling filter is used to generate luma inter prediction samples of non-affine inter modes.

3. The method of claim 2, wherein deriving, at the decoder, the third down-sampling filter comprises:

obtaining, at the decoder, first and last filter coefficients of the third and fourth down-sampling filters by merging first two filter coefficients and last two filter coefficients of the first and second down-sampling filters into two single filter coefficients.

4. The method of claim 1, wherein obtaining, at the decoder, the third down-sampling filter comprises:

obtaining, at the decoder, the third down-sampling filter based on applying a cosine windowed sinc function on top of scalable HEVC test model (SHM) filters.

5. The method of claim 4, wherein obtaining, at the decoder, the third down-sampling filter comprises:

obtaining, at the decoder, a frequency response of an ideal low-pass filter based on a cut-off frequency and scaling ratio;

obtaining, at the decoder, a cosine window function based on a filter length; and

obtaining, at the decoder, the third down-sampling filter based on the frequency response and the cosine window function.

6. The method of claim 5, wherein the cut-off frequency is equal to 0.9, the filter length is equal to 6, and the scaling ratio is equal to 1.5.

7. The method of claim 5, wherein the cut-off frequency is equal to 0.9, the filter length is equal to 6, and the scaling ratio is equal to 2.

8. The method of claim 1, wherein the fourth down-sampling filter is equal to the second down-sampling filter, wherein the fourth down-sampling filter is used to generate the chroma inter prediction samples of affine mode, and wherein the second down-sampling filter is used to generate the chroma inter prediction samples of non-affine mode.

9. The method of claim 8, wherein the fourth down-sampling filter is equal to an interpolation filter used to generate the chroma inter prediction samples of regular inter blocks when the resolution of the reference picture *I* is equal to that of the current picture.

10. The method of claim 8, wherein obtaining, at the decoder, the fourth down-sampling filter comprises:

obtaining, at the decoder, a frequency response of an ideal low-pass filter based on a cut-off frequency and scaling ratio;

obtaining, at the decoder, a cosine window function based on a filter length; and

obtaining, at the decoder, the fourth down-sampling filter based on the frequency

response and the cosine window function.

11. The method of claim 10, wherein the cut-off frequency is equal to 0.9 and the scaling ratio is equal to 1.5.

12. The method of claim 10, wherein the cut-off frequency is equal to 0.9 and the scaling ratio is equal to 2.

13. A computing device, comprising:

one or more processors; and

a non-transitory computer-readable storage medium storing instructions executable by the one or more processors, wherein the one or more processors are configured to:

obtain a reference picture I associated with a video block within the video signal;

obtain reference samples $I(i, j)$ of the video block from a reference block in the reference picture I, wherein i and j represent a coordinate of one sample within the video block;

obtain a first down-sampling filter and a second down-sampling filter to respectively generate luma and chroma inter prediction samples of the video block when the video block is coded in a non-affine inter mode and a resolution of the reference picture I is larger than that of a current picture;

obtain a third down-sampling filter and a fourth down-sampling filter to respectively generate the luma and chroma inter prediction samples of the video block when the video block is coded by affine mode and the resolution of the reference picture is larger than that of the current picture; and

obtain inter prediction samples of the video block based on the third and fourth down-sampling filters being applied to the reference samples $I(i, j)$.

14. The computing device of claim 13, wherein the one or more processors configured to obtain the third down-sampling filter and a fourth down-sampling filter are further configured to:

derive the third down-sampling filter is derived from the first down-sampling filter, wherein the third down-sampling filter is used to generate luma inter prediction samples of

affine mode, and wherein the first down-sampling filter is used to generate luma inter prediction samples of non-affine inter modes.

15. The computing device of claim 14, wherein the one or more processors configured to derive the third down-sampling filter are further configured to:

obtain first and last filter coefficients of the third and fourth down-sampling filters by merging first two filter coefficients and last two filter coefficients of the first and second down-sampling filters into two single filter coefficients.

16. The computing device of claim 13, wherein the one or more processors configured to obtain the third down-sampling filter are further configured to:

obtain the third down-sampling filter based on applying a cosine windowed sinc function on top of scalable HEVC test model (SHM) filters.

17. The computing device of claim 16, wherein the one or more processors configured to obtain the third down-sampling filter are further configured to:

obtain a frequency response of an ideal low-pass filter based on a cut-off frequency and scaling ratio;

obtain a cosine window function based on a filter length; and

obtain the third down-sampling filter based on the frequency response and the cosine window function.

18. The computing device of claim 17, wherein the cut-off frequency is equal to 0.9, the filter length is equal to 6, and the scaling ratio is equal to 1.5.

19. The computing device of claim 17, wherein the cut-off frequency is equal to 0.9, the filter length is equal to 6, and the scaling ratio is equal to 2.

20. The computing device of claim 13, wherein the fourth down-sampling filter is equal to the second down-sampling filter, wherein the fourth down-sampling filter is used to generate the chroma inter prediction samples of affine mode, and wherein the second down-sampling filter is used to generate the chroma inter prediction samples of non-affine mode.

21. The computing device of claim 20, wherein the fourth down-sampling filter is equal to an interpolation filter used to generate the chroma inter prediction samples of regular inter blocks when the resolution of the reference picture *I* is equal to that of the current picture.

22. The computing device of claim 20, wherein the one or more processors configured to obtain the fourth down-sampling filter are further configured to:

obtain a frequency response of an ideal low-pass filter based on a cut-off frequency and scaling ratio;

obtain a cosine window function based on a filter length; and

obtain the fourth down-sampling filter based on the frequency response and the cosine window function.

23. The computing device of claim 22, wherein the cut-off frequency is equal to 0.9 and the scaling ratio is equal to 1.5.

24. The computing device of claim 22, wherein the cut-off frequency is equal to 0.9 and the scaling ratio is equal to 2.

25. A non-transitory computer-readable storage medium storing a plurality of programs for execution by a computing device having one or more processors, wherein the plurality of programs, when executed by the one or more processors, cause the computing device to perform acts comprising:

obtaining, at a decoder, a reference picture *I* associated with a video block within the video signal;

obtaining, at the decoder, reference samples $I(i, j)$ of the video block from a reference block in the reference picture *I*, wherein i and j represent a coordinate of one sample within the video block;

obtaining, at the decoder, a first down-sampling filter and a second down-sampling filter to respectively generate luma and chroma inter prediction samples of the video block when the video block is coded in a non-affine inter mode and a resolution of the reference picture *I* is larger than that of a current picture;

obtaining, at the decoder, a third down-sampling filter and a fourth down-sampling

filter to respectively generate the luma and chroma inter prediction samples of the video block when the video block is coded by affine mode and the resolution of the reference picture is larger than that of the current picture; and

obtaining, at the decoder, inter prediction samples of the video block based on the third and fourth down-sampling filters being applied to the reference samples $I(i,j)$.

26. The non-transitory computer-readable storage medium of claim 25, wherein the plurality of programs further cause the computing device to perform:

deriving, at the decoder, the third down-sampling filter is derived from the first down-sampling filter, wherein the third down-sampling filter is used to generate luma inter prediction samples of affine mode, and wherein the first down-sampling filter is used to generate luma inter prediction samples of non-affine inter modes.

27. The non-transitory computer-readable storage medium of claim 26, wherein the plurality of programs further cause the computing device to perform:

obtaining, at the decoder, first and last filter coefficients of the third and fourth down-sampling filters by merging first two filter coefficients and last two filter coefficients of the first and second down-sampling filters into two single filter coefficients.

28. The non-transitory computer-readable storage medium of claim 25, wherein the plurality of programs further cause the computing device to perform:

obtaining, at the decoder, the third down-sampling filter based on applying a cosine windowed sinc function on top of scalable HEVC test model (SHM) filters.

29. The non-transitory computer-readable storage medium of claim 28, wherein the plurality of programs further cause the computing device to perform:

obtaining, at the decoder, a frequency response of an ideal low-pass filter based on a cut-off frequency and scaling ratio;

obtaining, at the decoder, a cosine window function based on a filter length; and

obtaining, at the decoder, the third down-sampling filter based on the frequency response and the cosine window function.

30. The non-transitory computer-readable storage medium of claim 29, wherein the cut-off frequency is equal to 0.9, the filter length is equal to 6, and the scaling ratio is equal to 1.5.

31. The non-transitory computer-readable storage medium of claim 29, wherein the cut-off frequency is equal to 0.9, the filter length is equal to 6, and the scaling ratio is equal to 2.

32. The non-transitory computer-readable storage medium of claim 25, wherein the fourth down-sampling filter is equal to the second down-sampling filter, wherein the fourth down-sampling filter is used to generate the chroma inter prediction samples of affine mode, and wherein the second down-sampling filter is used to generate the chroma inter prediction samples of non-affine mode.

33. The non-transitory computer-readable storage medium of claim 32, wherein the fourth down-sampling filter is equal to an interpolation filter used to generate the chroma inter prediction samples of regular inter blocks when the resolution of the reference picture *I* is equal to that of the current picture.

34. The non-transitory computer-readable storage medium of claim 32, wherein the plurality of programs further cause the computing device to perform:

obtaining, at the decoder, a frequency response of an ideal low-pass filter based on a cut-off frequency and scaling ratio;

obtaining, at the decoder, a cosine window function based on a filter length; and

obtaining, at the decoder, the fourth down-sampling filter based on the frequency response and the cosine window function.

35. The non-transitory computer-readable storage medium of claim 34, wherein the cut-off frequency is equal to 0.9 and the scaling ratio is equal to 1.5.

36. The non-transitory computer-readable storage medium of claim 34, wherein the cut-off frequency is equal to 0.9 and the scaling ratio is equal to 2.

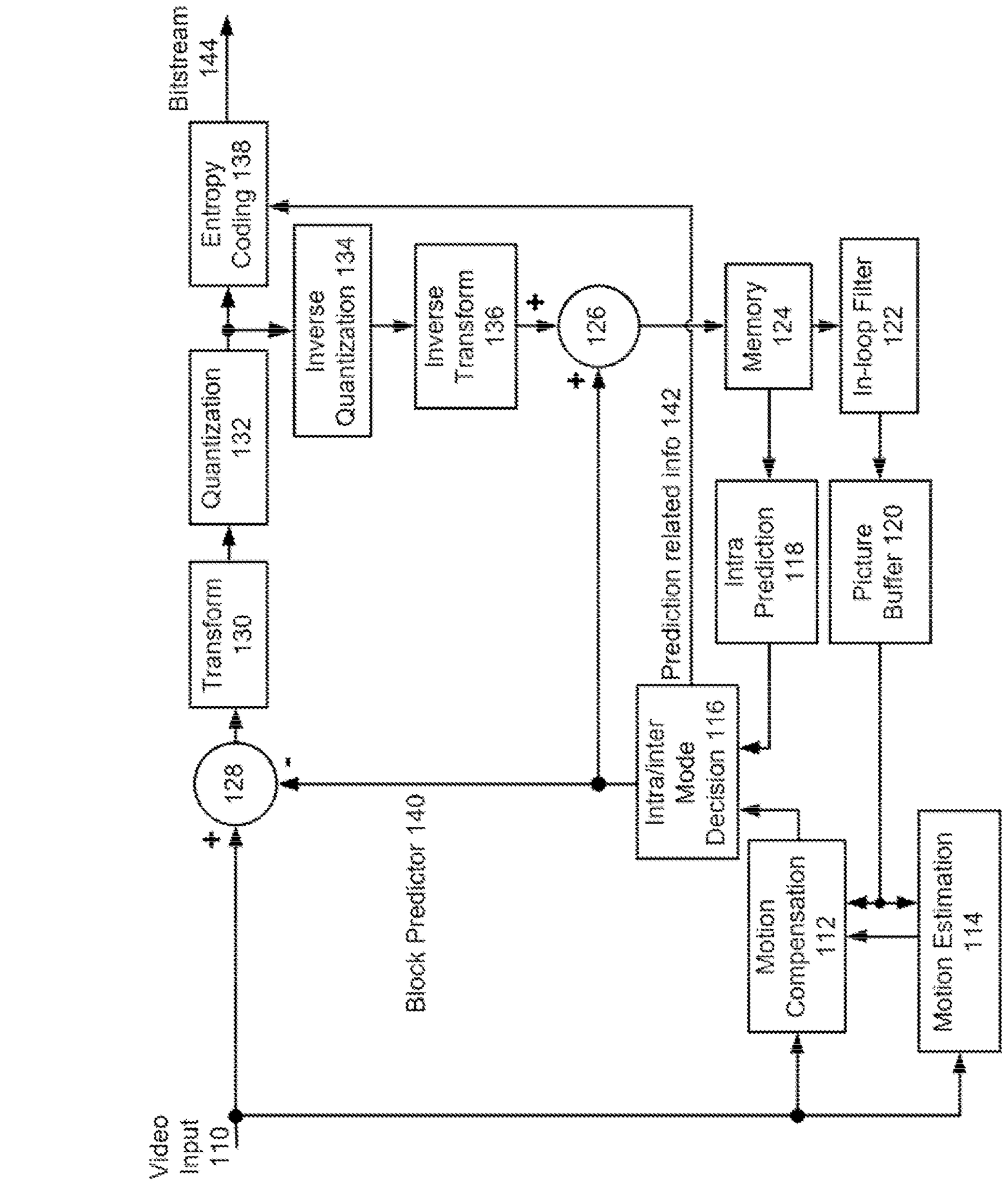

Bitstream 144
Entropy Coding 138
Quantization 132
Transform 130
Inverse Quantization 134
Inverse Transform 136
126
Prediction related info 142
Memory 124
In-loop Filter 122
Intra Prediction 118
Picture Buffer 120
Intra/inter Mode Decision 116
Motion Compensation 112
Motion Estimation 114
128
Block Predictor 140
Video Input 110
100

FIG. 1

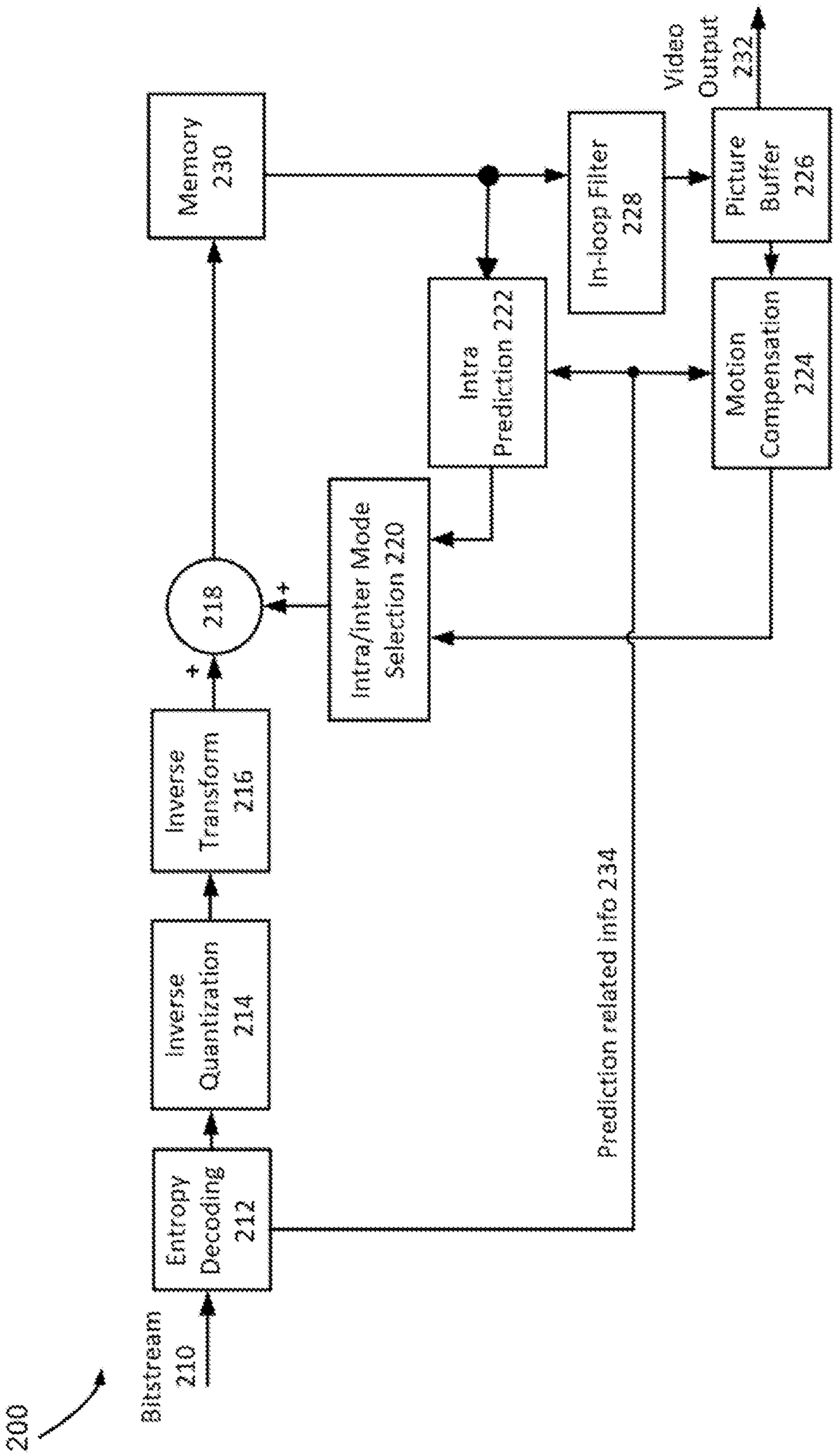

Memory 230
Video Output 232
In-loop Filter 228
Picture Buffer 226
Intra Prediction 222
Motion Compensation 224
Intra/inter Mode Selection 220
218
Inverse Transform 216
Inverse Quantization 214
Entropy Decoding 212
Prediction related info 234
Bitstream 210
200

FIG. 2

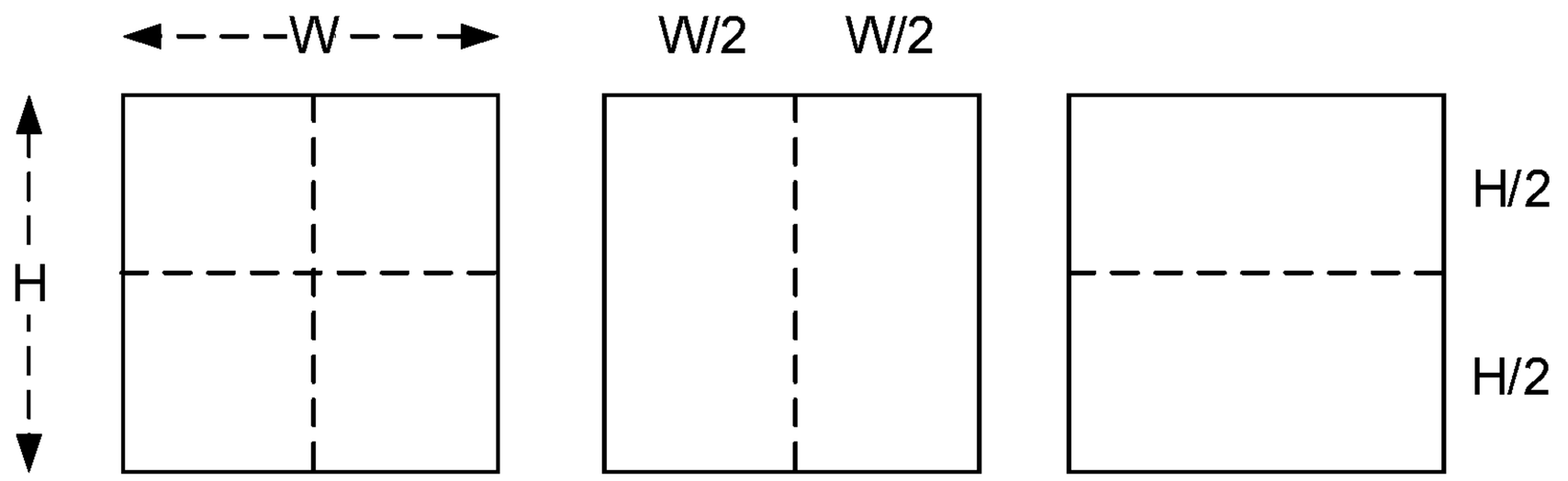


FIG. 3A

FIG. 3B

FIG. 3C

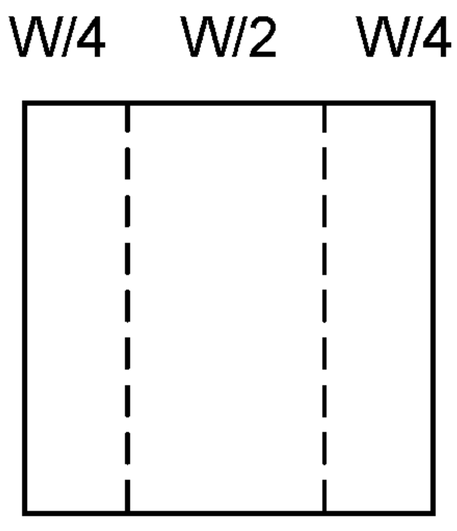


FIG. 3D

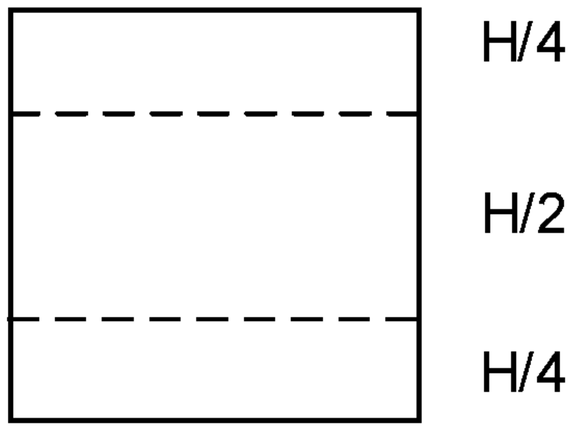


FIG. 3E

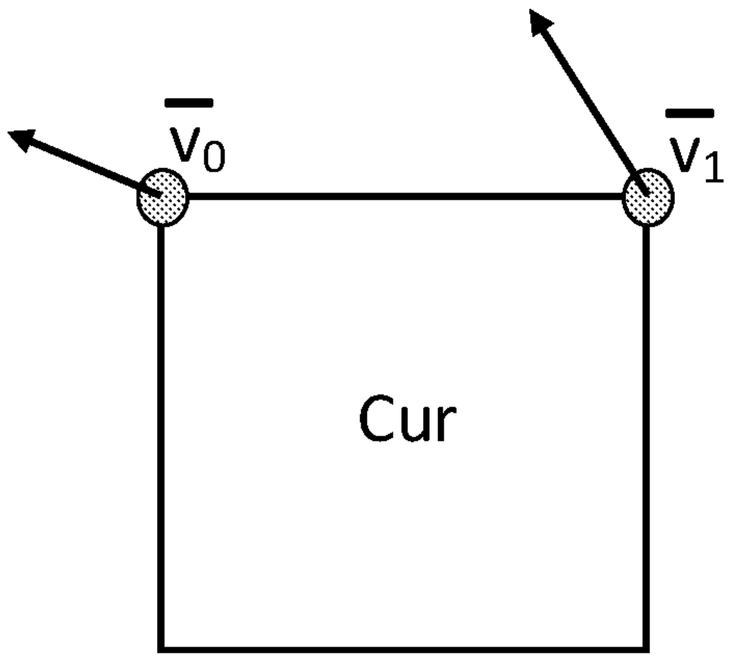


FIG. 4A

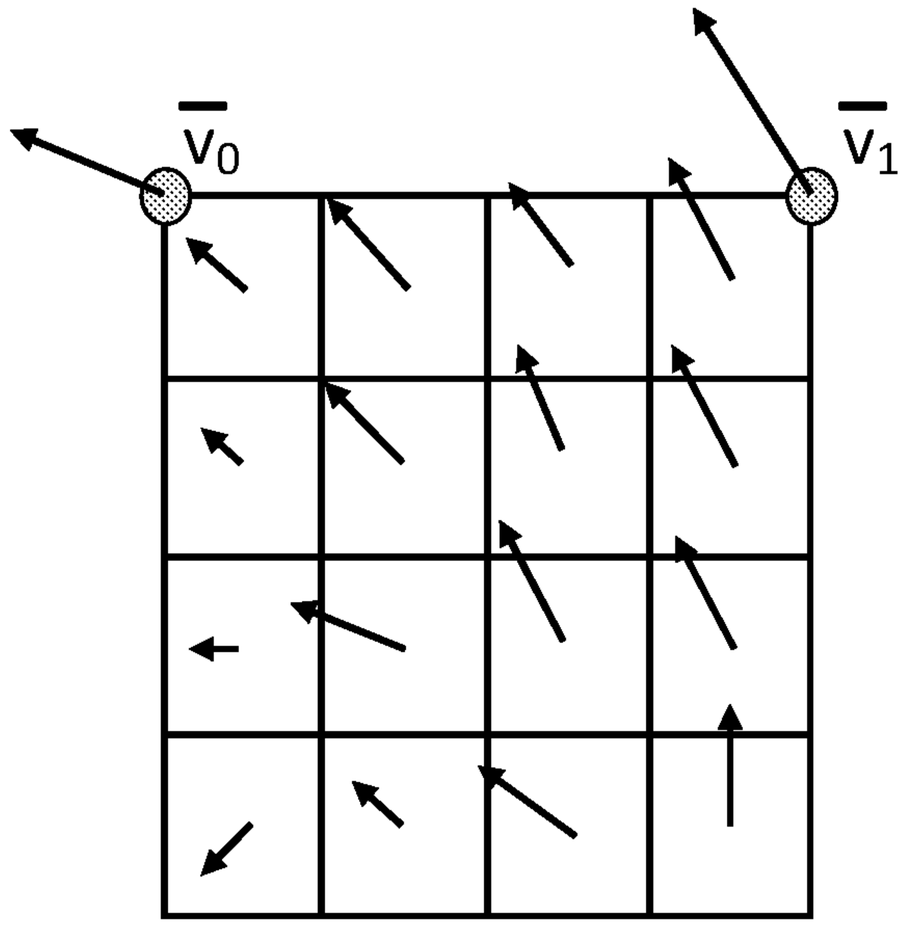


FIG. 4B

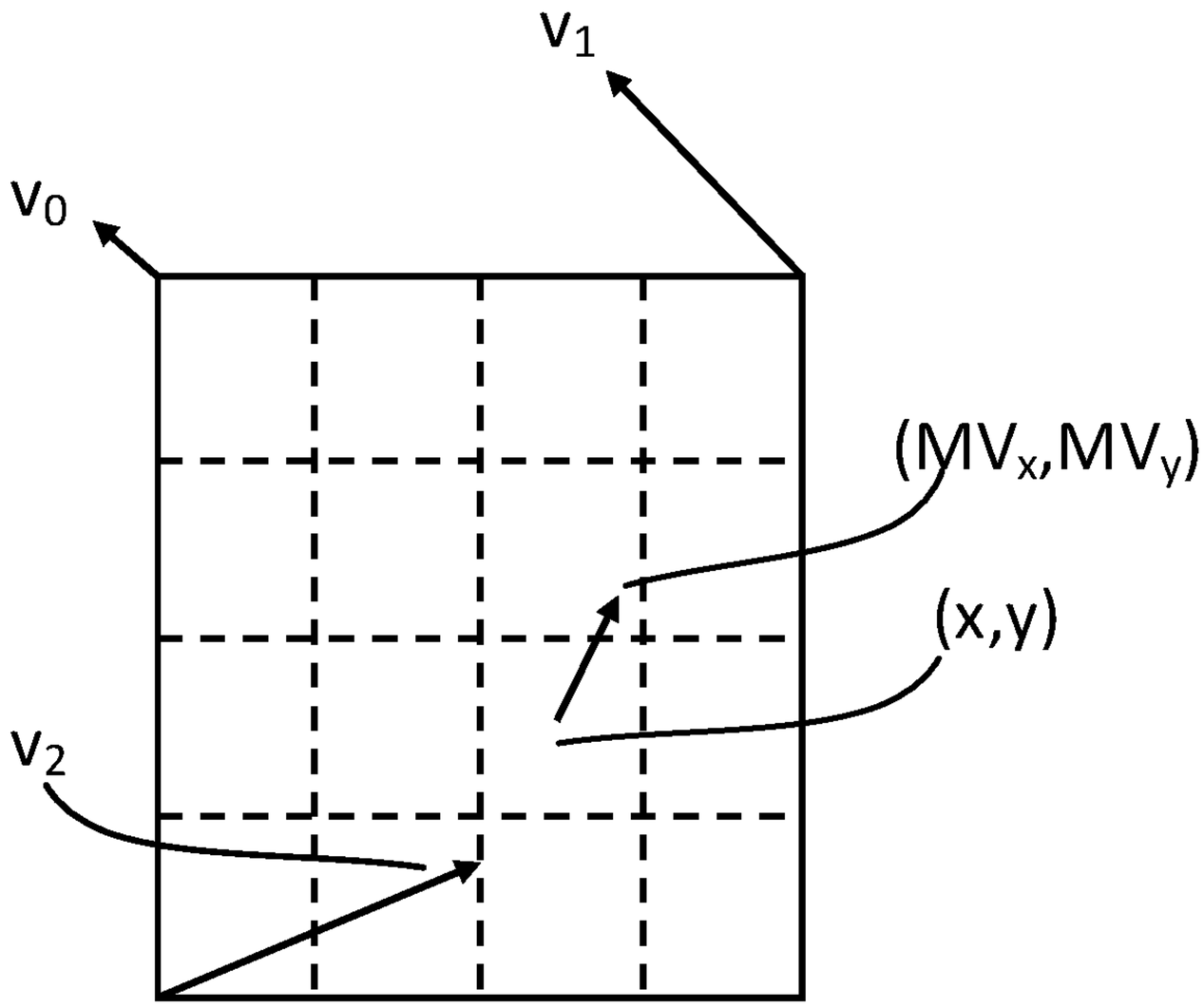

v_1
v_0
(MV_x,MV_y)
(x,y)
v_2

FIG. 5

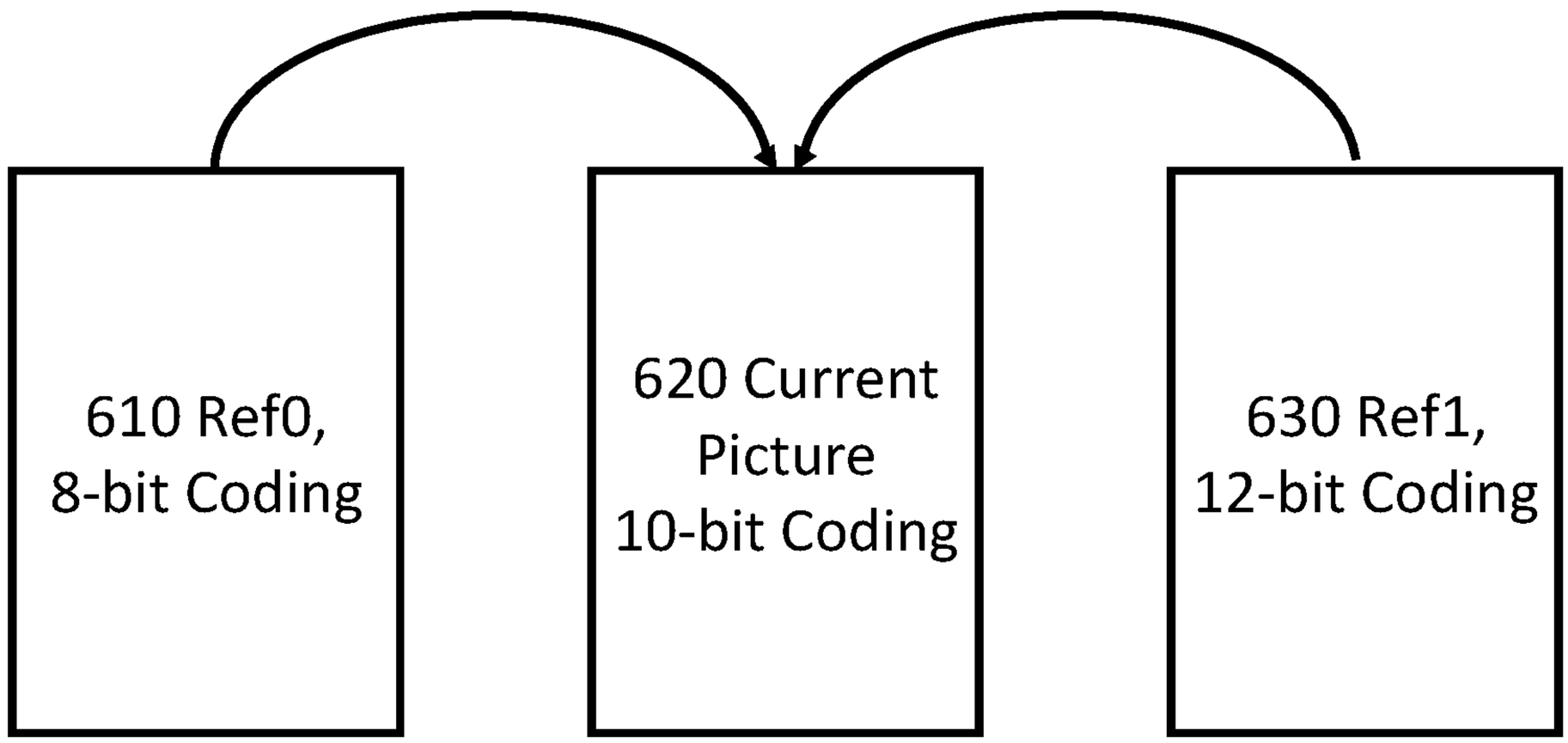

610 Ref0,
8-bit Coding
620 Current
Picture
10-bit Coding
630 Ref1,
12-bit Coding

FIG. 6

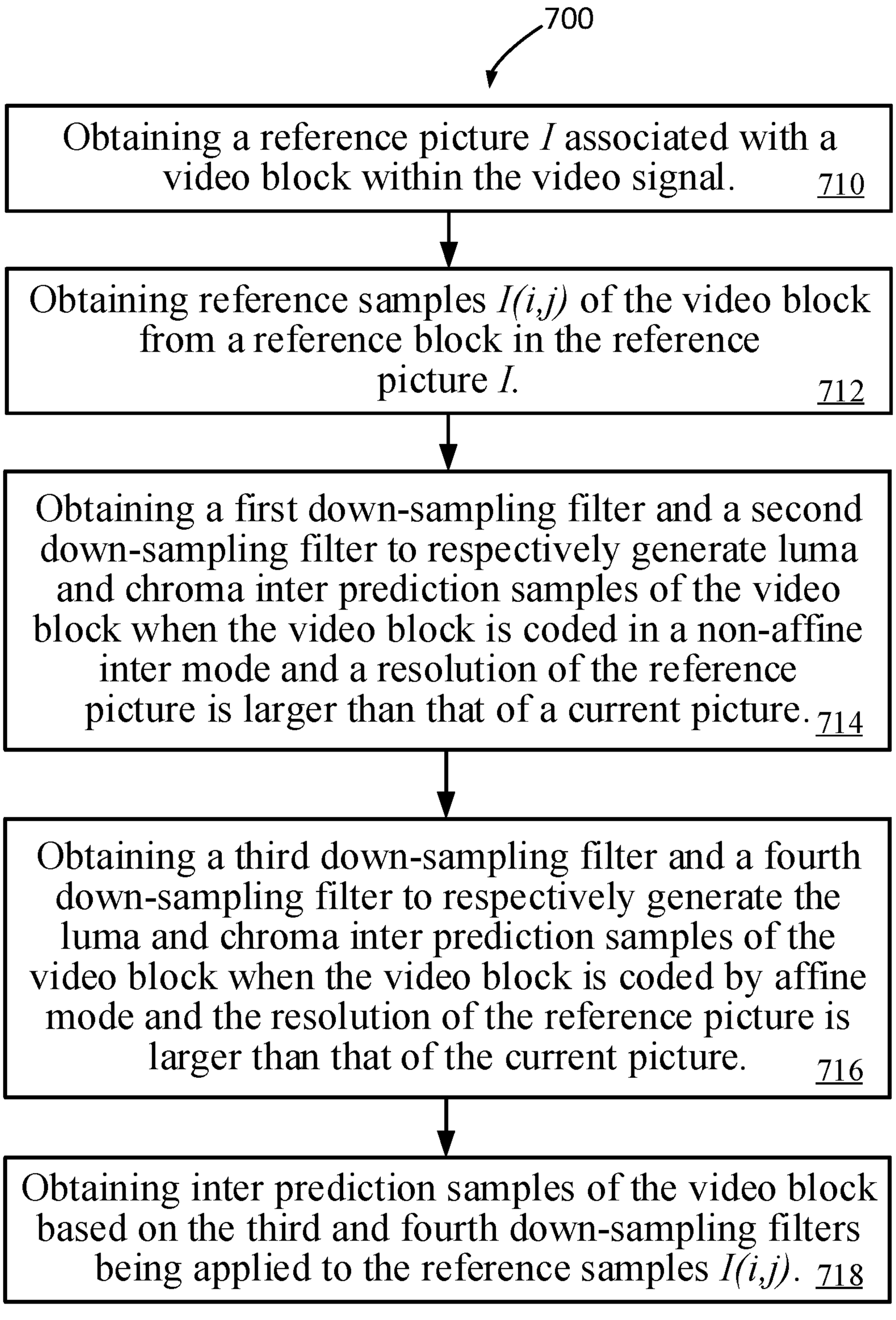

700
Obtaining a reference picture *I* associated with a video block within the video signal. 710
Obtaining reference samples *I(i,j)* of the video block from a reference block in the reference picture *I*. 712
Obtaining a first down-sampling filter and a second down-sampling filter to respectively generate luma and chroma inter prediction samples of the video block when the video block is coded in a non-affine inter mode and a resolution of the reference picture is larger than that of a current picture. 714
Obtaining a third down-sampling filter and a fourth down-sampling filter to respectively generate the luma and chroma inter prediction samples of the video block when the video block is coded by affine mode and the resolution of the reference picture is larger than that of the current picture. 716
Obtaining inter prediction samples of the video block based on the third and fourth down-sampling filters being applied to the reference samples *I(i,j)*. 718

FIG. 7

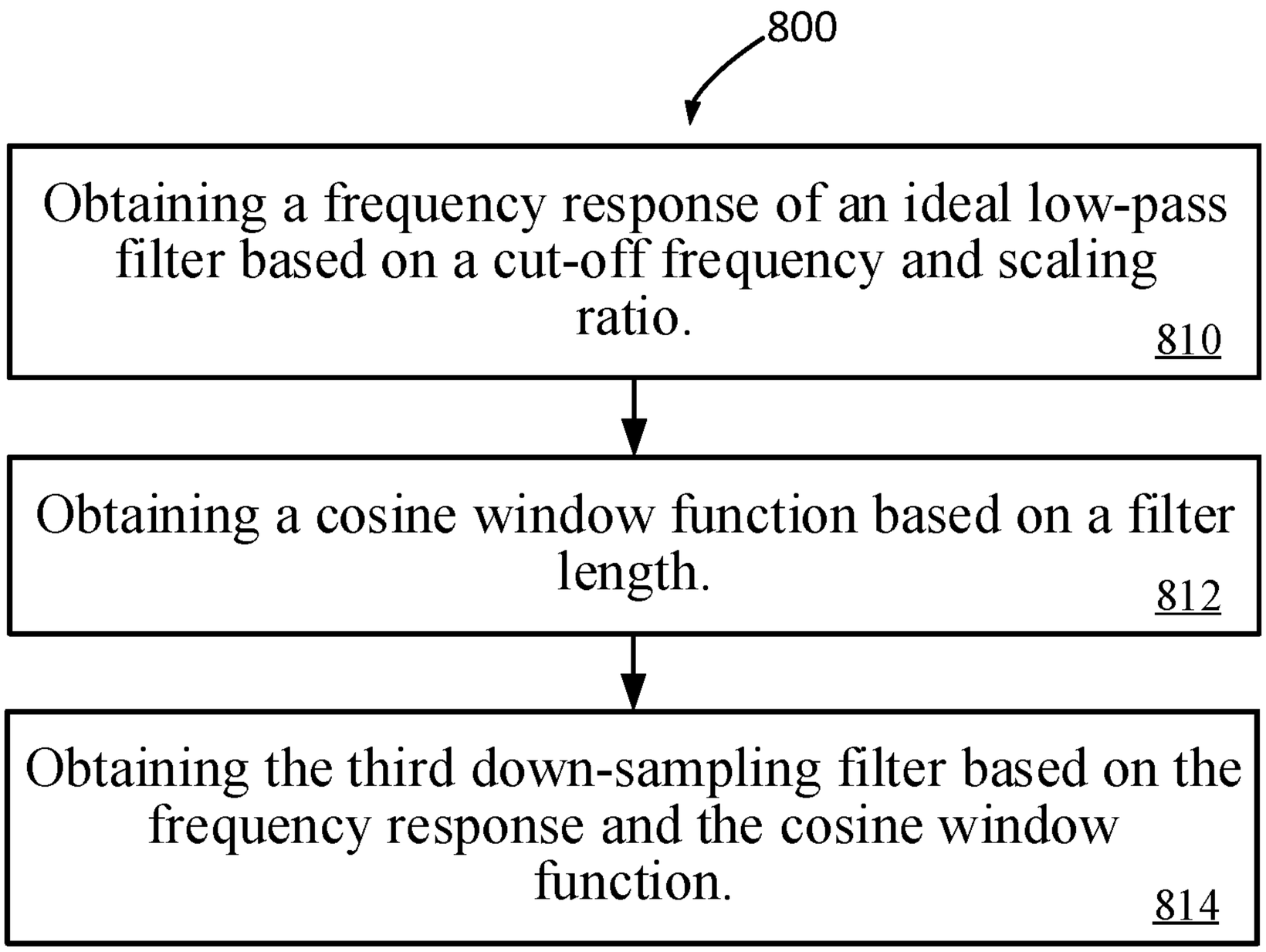

800
Obtaining a frequency response of an ideal low-pass filter based on a cut-off frequency and scaling ratio.
810
Obtaining a cosine window function based on a filter length.
812
Obtaining the third down-sampling filter based on the frequency response and the cosine window function.
814

FIG. 8

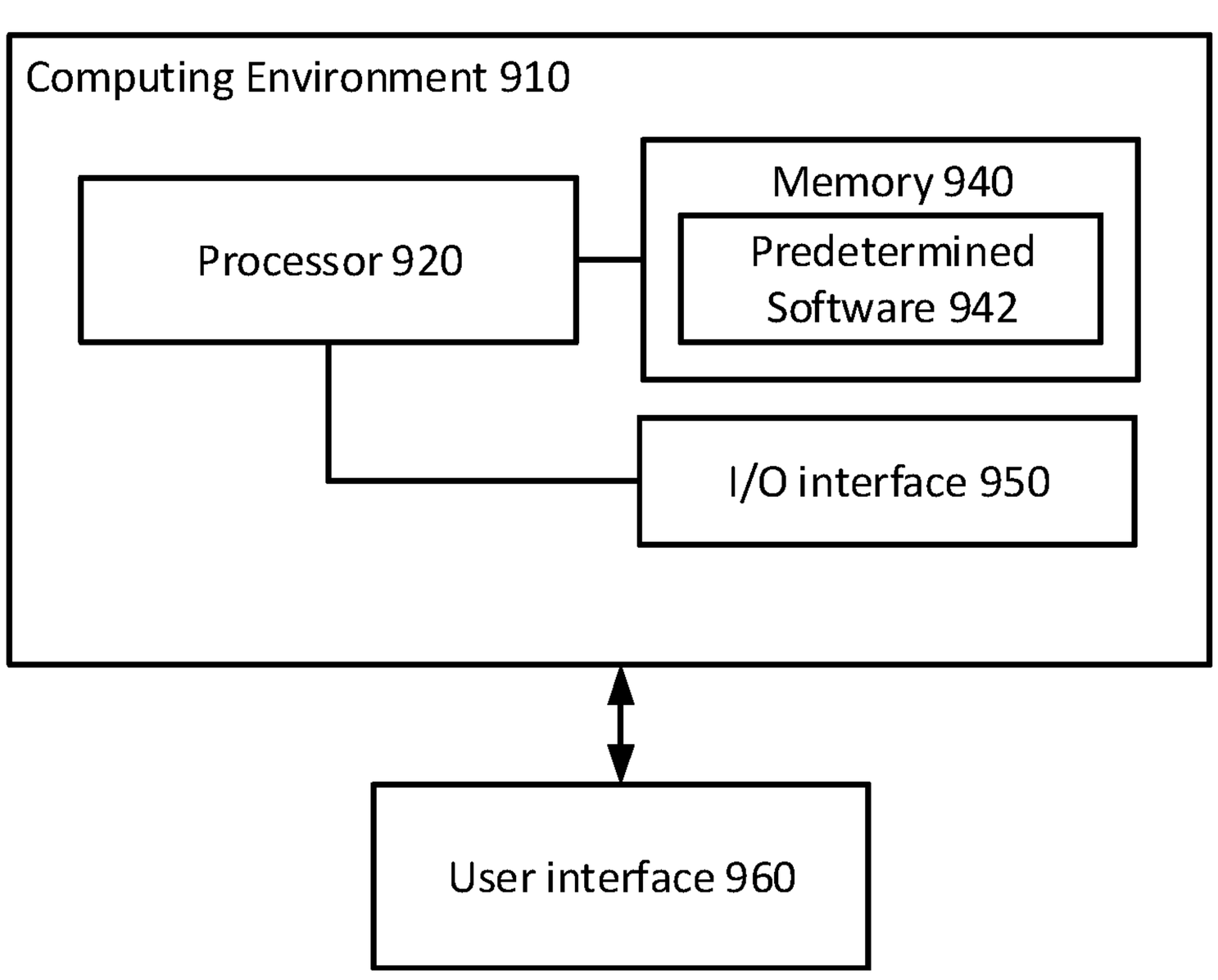

Computing Environment 910
Processor 920
Memory 940
Predetermined Software 942
I/O interface 950
User interface 960

FIG. 9

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2020/067017

A. CLASSIFICATION OF SUBJECT MATTER

H04N 19/132(2014.01)i; **H04N 19/80**(2014.01)i; **H04N 19/109**(2014.01)i; **H04N 19/105**(2014.01)i; **H04N 19/186**(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 19/132(2014.01)

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: reference picture resampling, down-sampling, filter, affine

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	J. SAMUELSSON et al., `AHG 8: Adaptive Resolution Change (ARC) with downsampling`, JVET-O0240-v1, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 15th Meeting: Gothenburg, SE, 05 July 2019 pages 1-8	1-36
Y	JONATAN SAMUELSSON et al., `CE1-1: RPR downsampling filter`, JVET-P0088, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 16th Meeting: Geneva, CH, 10 October 2019 pages 1-4	1-36
Y	JIANCONG LUO et al., `CE1-related: Reference picture resampling filters`, JVET-P0353-v5, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 16th Meeting: Geneva, CH, 10 October 2019 pages 1-5	2-7,10-12,14-19, 22-24,26-31,34-36
A	DANIEL LUO et al., `Description of Core Experiment 1 (CE1): Reference picture resampling filters`, JVET-O2021-v3, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/ WG 11, 15th Meeting: Gothenburg, SE, 11 September 2019 pages 1-2	1-36

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

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"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
16 April 2021	**19 April 2021**
Name and mailing address of the ISA/KR **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/US2020/067017

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	PEISONG CHEN et al., `AHG 8: Adaptive Resolution Change`, JVET-O0303-v2, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 15th Meeting: Gothenburg, SE, 05 July 2019 pages 1-7	1-36