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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
01 April 2021 (01.04.2021)

WIPO | PCT

(10) International Publication Number
WO 2021/061312 A1

(51) International Patent Classification:
H04N 19/59 (2014.01) *H04N 19/517* (2014.01)
H04N 19/80 (2014.01) *H04N 19/51* (2014.01)
H04N 19/117 (2014.01) *H04N 19/42* (2014.01)
H04N 19/176 (2014.01) *H04N 19/44* (2014.01)

(21) International Application Number:
PCT/US2020/047249

(22) International Filing Date:
20 August 2020 (20.08.2020)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/904,608 23 September 2019 (23.09.2019) US
62/906,930 27 September 2019 (27.09.2019) US

(71) Applicant: **ALIBABA GROUP HOLDING LIMITED** [—/US]; Fourth Floor, One Capital Place, P.O. Box 847, George Town, Grand Cayman (KY).

(72) Inventors: **LUO, Jiancong**; Alibaba Group Legal Department, 400 South El Camino Real, Suite 400, San Mateo, CA 94402 (US). **YE, Yan**; Alibaba Group Legal Department, 400 South El Camino Real, Suite 400, San Mateo, CA 94402 (US). **SARWER, Mohammed, Golam**; Alibaba Group Legal Department, 400 South El Camino Real, Suite 400, San Mateo, CA 94402 (US).

(74) Agent: **CAPRON, Aaron, J.**; Finnegan, Henderson, Farabow, Garrett & Dunner LLP, 901 New York Avenue, NW, Washington, DC 2000-14413 (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,

WO 2021/061312 A1

(54) Title: FILTERS FOR MOTION COMPENSATION INTERPOLATION WITH REFERENCE DOWN-SAMPLING

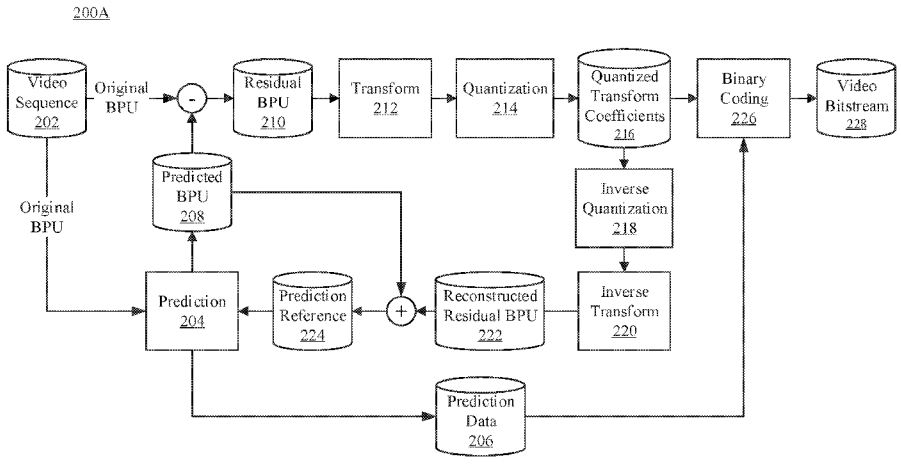


FIG. 2A

(57) Abstract: The present disclosure provides systems and methods for processing video content using motion compensation interpolation. The methods include: in response to a target picture and a reference picture having different resolutions, applying a band-pass filter to the reference picture, to perform a motion compensated interpolation with reference down- sampling to generate a reference block; and encoding or decoding a block of the target picture using the reference block.

[Continued on next page]

SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

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

FILTERS FOR MOTION COMPENSATION INTERPOLATION WITH REFERENCE DOWN-SAMPLING

CROSS REFERENCE TO RELATED APPLICATIONS

[001] The disclosure claims the benefits of priority to U.S. Provisional Application No. 62/904,608, filed September 23, 2019, and U.S. Provisional Application 62/906,930, filed September 27, 2019, both of which are incorporated herein by reference in their entireties.

TECHNICAL FIELD

[002] The present disclosure generally relates to video processing, and more particularly, to filters for motion compensation interpolation with reference resampling.

BACKGROUND

[003] A video is a set of static pictures (or "frames") capturing the visual information. To reduce the storage memory and the transmission bandwidth, a video can be compressed before storage or transmission and decompressed before display. The compression process is usually referred to as encoding and the decompression process is usually referred to as decoding. There are various video coding formats which use standardized video coding technologies, most commonly based on prediction, transform, quantization, entropy coding and in-loop filtering. The video coding standards, such as the High Efficiency Video Coding (HEVC/H.265) standard, the Versatile Video Coding (VVC/H.266) standard, AVS standards, specifying the specific video coding formats, are developed by standardization organizations. With more and more advanced video coding technologies being adopted in the video standards, the coding efficiency of the new video coding standards get higher and higher.

SUMMARY OF THE DISCLOSURE

[004] Embodiments of the disclosure provide a computer-implemented method for performing motion compensation interpolation with reference resampling. The method can include: in response to a target picture and a reference picture having different resolutions, applying a band-pass filter to the reference picture, to perform a motion compensated interpolation with reference down-sampling to generate a reference block; and processing a block of the target picture using the reference block.

[005] Embodiments of the disclosure further provide a system for performing motion compensation interpolation. The system can include: a memory storing a set of instructions; and at least one processor configured to execute set of instructions to cause the device to perform: in response to a target picture and a reference picture having different resolutions, applying a band-pass filter to the reference picture, to perform a motion compensated interpolation with reference down-sampling to generate a reference block; and processing a block of the target picture using the reference block.

[006] Embodiments of the disclosure further also provide a non-transitory computer readable medium that stores a set of instructions that is executable by at least one processor of a computer system to cause the computer system to perform a method for processing video content. Thee method can include: in response to a target picture and a reference picture having different resolutions, applying a band-pass filter to the reference picture, to perform a motion compensated interpolation with reference down-sampling to generate a reference block; and processing a block of the target picture using the reference block.

BRIEF DESCRIPTION OF THE DRAWINGS

[007] Embodiments and various aspects of the present disclosure are illustrated in the following detailed description and the accompanying figures. Various features shown in the figures are not drawn to scale.

[008] **FIG. 1** illustrates structures of an exemplary video sequence, consistent with embodiments of the disclosure, consistent with embodiments of the disclosure.

[009] **FIG. 2A** illustrates a schematic diagram of an exemplary encoding process of a hybrid video coding system, consistent with embodiments of the disclosure.

[010] **FIG. 2B** illustrates a schematic diagram of another exemplary encoding process of a hybrid video coding system, consistent with embodiments of the disclosure.

[011] **FIG. 3A** illustrates a schematic diagram of an exemplary decoding process of a hybrid video coding system, consistent with embodiments of the disclosure.

[012] **FIG. 3B** illustrates a schematic diagram of another exemplary decoding process of a hybrid video coding system, consistent with embodiments of the disclosure.

[013] **FIG. 4** is a block diagram of an exemplary apparatus for encoding or decoding a video, consistent with embodiments of the disclosure.

[014] **FIG. 5** illustrates a schematic diagram of reference pictures and a current picture, consistent with embodiments of the disclosure.

[015] **FIG. 6** illustrates a table of exemplary 6-tap DCT-based interpolation filter coefficients for a 4×4 luma block, consistent with embodiments of the disclosure.

[016] **FIG. 7** illustrates a table of exemplary 8-tap interpolation filter coefficients for luma components, consistent with embodiments of the disclosure.

[017] **FIG. 8** illustrates a table of exemplary 4-tap 32-phase interpolation filter coefficients for chroma components, consistent with embodiments of the disclosure.

[018] **FIG. 9** illustrates frequency response of an exemplary ideal low pass filter, consistent with embodiments of the disclosure.

[019] **FIG. 10** illustrates a table of exemplary 12-tap cosine-windowed-sinc interpolation filter coefficients for 2:1 down-sampling, consistent with embodiments of the disclosure.

[020] **FIG. 11** illustrates a table of exemplary 12-tap cosine-windowed-sinc interpolation filter coefficients for 1.5:1 down-sampling, consistent with embodiments of the disclosure.

[021] **FIG. 12** illustrates an exemplary luma sample interpolation filtering process for reference down-sampling, consistent with embodiments of the disclosure.

[022] **FIG. 13** illustrates an exemplary chroma sample interpolation filtering process for reference down-sampling, consistent with embodiments of the disclosure.

[001] **FIG. 14** illustrates an exemplary luma sample interpolation filtering process for reference down-sampling, consistent with embodiments of the disclosure.

[002] **FIG. 15** illustrates an exemplary chroma sample interpolation filtering process for reference down-sampling, consistent with embodiments of the disclosure.

[003] **FIG. 16** illustrates an exemplary luma sample interpolation filtering process for reference down-sampling, consistent with embodiments of the disclosure.

[004] **FIG. 17** illustrates an exemplary chroma sample interpolation filtering process for reference down-sampling, consistent with embodiments of the disclosure.

[005] **FIG. 18** illustrates an exemplary chroma sample interpolation filtering process for reference down-sampling, consistent with embodiments of the disclosure.

[006] **FIG. 19** illustrates an exemplary 8-tap filter for MC interpolation with reference down-sampling at ratio 2:1, consistent with embodiments of the disclosure.

[007] **FIG. 20** illustrates an exemplary 8-tap filter for MC interpolation with reference down-sampling at ratio 1.5:1, consistent with embodiments of the disclosure.

[008] **FIG. 21** illustrates an exemplary 8-tap filter for MC interpolation with reference down-sampling at ratio 2:1 with 32 phases, consistent with embodiments of the disclosure.

[009] **FIG. 22** illustrates an exemplary 8-tap filter for MC interpolation with reference down-sampling at ratio 1.5:1 with 32 phases, consistent with embodiments of the disclosure.

[010] **FIG. 23** illustrates a table of exemplary 6-tap filter coefficients for 4×4 luma block MC interpolation with reference down-sampling ratio 2:1 with 16 phases, consistent with embodiments of the disclosure.

[011] **FIG. 24** illustrates a table of exemplary 6-tap filter coefficients for luma 4×4 block MC interpolation with reference down-sampling ratio 1.5:1 with 16 phases, consistent with embodiments of the disclosure.

[012] **FIG. 25** illustrates a table of an exemplary 8-tap filter for MC interpolation with reference down-sampling at ratio 2:1, consistent with embodiments of the disclosure.

[013] **FIG. 26** illustrates an exemplary 8-tap filter for MC interpolation with reference down-sampling at ratio 1.5:1, consistent with embodiments of the disclosure.

[014] **FIG. 27** illustrates an exemplary 6-tap filter for MC interpolation with reference down-sampling at ratio 2:1 with 16 phases, consistent with embodiments of the disclosure.

[015] **FIG. 28** illustrates an exemplary 6-tap filter for MC interpolation with reference down-sampling at ratio 1.5:1 with 16 phases, consistent with embodiments of the disclosure.

[016] **FIG. 29** illustrates an exemplary 4-tap filter for MC interpolation with reference down-sampling at ratio 2:1 with 32-phases, consistent with embodiments of the disclosure.

[017] **FIG. 30** illustrates an exemplary 4-tap filter for MC interpolation with reference down-sampling at ratio 1.5:1 with 32-phases, consistent with embodiments of the disclosure.

[018] **FIG. 31** illustrates an exemplary 8-tap filter for MC interpolation with reference down-sampling at ratio 2:1, consistent with embodiments of the disclosure.

[019] **FIG. 32** illustrates an exemplary 8-tap filter for MC interpolation with reference down-sampling at ratio 1.5:1, consistent with embodiments of the disclosure.

[020] **FIG. 33** illustrates an exemplary 6-tap filter for MC interpolation with reference down-sampling at ratio 2:1 with 16 phases, consistent with embodiments of the disclosure.

[021] **FIG. 34** illustrates an exemplary 6-tap filter for MC interpolation with reference down-sampling at ratio 1.5:1 with 16 phases, consistent with embodiments of the disclosure.

[022] **FIG. 35** illustrates an exemplary 4-tap filter for MC interpolation with reference downsampling at ratio 2:1 with 32-phases, consistent with embodiments of the disclosure.

[023] **FIG. 36** illustrates an exemplary 4-tap filter for MC interpolation with reference downsampling at ratio 1.5:1 with 32-phases, consistent with embodiments of the disclosure.

[024] **FIG. 37** illustrates an example of filter coefficient signaling, consistent with embodiments of the disclosure.

[025] **FIG. 38** illustrates an exemplary syntax structure to signal the resampling ratio and corresponding filter set, consistent with embodiments of the disclosure.

[026] **FIG. 39** is a flowchart of an exemplary method for processing video content, consistent with embodiments of the disclosure.

[027] **FIG. 40** is a flowchart of an exemplary method for rounding filter coefficients of a cosine windowed-sinc filter, consistent with embodiments of the disclosure.

DETAILED DESCRIPTION

[028] Reference will now be made in detail to exemplary 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 exemplary embodiments do not represent all implementations consistent with the invention. Instead, they are merely examples of apparatuses and methods consistent with aspects related to the invention as recited in the appended claims. Particular aspects of the present disclosure are described in greater detail below. The terms and definitions provided herein control, if in conflict with terms and/or definitions incorporated by reference.

[029] Video coding systems are often used to compress digital video signals, for instance to reduce storage space consumed or to reduce transmission bandwidth consumption associated with such signals. With high-definition (HD) videos (e.g., having a resolution of 1920×1080 pixels) gaining popularity in various applications of video compression, such as online video streaming, video conferencing, or video monitoring, it is a continuous need to develop video coding tools that can increase compression efficiency of video data.

[030] For example, video monitoring applications are increasingly and extensively used in many application scenarios (e.g., security, traffic, environment monitoring, or the like), and the numbers and resolutions of the monitoring devices keep growing rapidly. Many video monitoring application scenarios prefer to provide HD videos to users to capture more information, which has more pixels per frame to capture such information. However, an HD video bitstream can have a high bitrate that demands high bandwidth for transmission and large space for storage. For example, a monitoring video stream having an average 1920×1080 resolution can require a bandwidth as high as 4 Mbps for real-time transmission.

Also, the video monitoring generally monitors 7×24 continuously, which can greatly challenge a storage system, if the video data is to be stored. The demand for high bandwidth and large storage of the HD videos has therefore become a major limitation to its large-scale deployment in video monitoring.

[031] A video is a set of static pictures (or "frames") arranged in a temporal sequence to store visual information. A video capture device (e.g., a camera) can be used to capture and store those pictures in a temporal sequence, and a video playback device (e.g., a television, a computer, a smartphone, a tablet computer, a video player, or any end-user terminal with a function of display) can be used to display such pictures in the temporal sequence. Also, in some applications, a video capturing device can transmit the captured video to the video playback device (e.g., a computer with a monitor) in real-time, such as for monitoring, conferencing, or live broadcasting.

[032] For reducing the storage space and the transmission bandwidth needed by such applications, the video can be compressed before storage and transmission and decompressed before the display. The compression and decompression can be implemented by software executed by a processor (e.g., a processor of a generic computer) or specialized hardware. The module for compression is generally referred to as an "encoder," and the module for decompression is generally referred to as a "decoder." The encoder and decoder can be collectively referred to as a "codec." The encoder and decoder can be implemented as any of a variety of suitable hardware, software, or a combination thereof. For example, the hardware implementation of the encoder and decoder can include circuitry, such as one or more microprocessors, digital signal processors (DSPs), application-specific integrated circuits (ASICs), field-programmable gate arrays (FPGAs), discrete logic, or any combinations thereof. The software implementation of the encoder and decoder can include program codes, computer-executable instructions, firmware, or any suitable computer-implemented algorithm

or process fixed in a computer-readable medium. Video compression and decompression can be implemented by various algorithms or standards, such as MPEG- 1, MPEG-2, MPEG-4, H.26x series, or the like. In some applications, the codec can decompress the video from a first coding standard and re-compress the decompressed video using a second coding standard, in which case the codec can be referred to as a "transcoder."

[033] The video encoding process can identify and keep useful information that can be used to reconstruct a picture and disregard unimportant information for the reconstruction. If the disregarded, unimportant information cannot be fully reconstructed, such an encoding process can be referred to as "lossy." Otherwise, it can be referred to as "lossless." Most encoding processes are lossy, which is a tradeoff to reduce the needed storage space and the transmission bandwidth.

[034] The useful information of a picture being encoded (referred to as a "current picture") include changes with respect to a reference picture (e.g., a picture previously encoded and reconstructed). Such changes can include position changes, luminosity changes, or color changes of the pixels, among which the position changes are mostly concerned. Position changes of a group of pixels that represent an object can reflect the motion of the object between the reference picture and the current picture.

[035] A picture coded without referencing another picture (i.e., it is its own reference picture) is referred to as an "I-picture." A picture coded using a previous picture as a reference picture is referred to as a "P-picture." A picture coded using both a previous picture and a future picture as reference pictures (i.e., the reference is "bi-directional") is referred to as a "B-picture."

[036] As previously mentioned, video monitoring that uses HD videos faces challenges of demands of high bandwidth and large storage. For addressing such challenges, the bitrate of the encoded video can be reduced. Among the I-, P-, and B-pictures, I-pictures

have the highest bitrate. Because the backgrounds of most monitoring videos are nearly static, one way to reduce the overall bitrate of the encoded video can be using fewer I-pictures for video encoding.

[037] However, the improvement of using fewer I-pictures can be trivial because the I-pictures are typically not dominant in the encoded video. For example, in a typical video bitstream, the ratio of I-, B-, and P-pictures can be 1:20:9, in which the I-pictures can account for less than 10% of the total bitrate. In other words, in such an example, even all I-pictures are removed, the reduced bitrate can be no more than 10%.

[038] **FIG. 1** illustrates structures of an example video sequence 100, consistent with embodiments of the disclosure. Video sequence 100 can be a live video or a video having been captured and archived. Video 100 can be a real-life video, a computer-generated video (e.g., computer game video), or a combination thereof (e.g., a real-life video with augmented-reality effects). Video sequence 100 can be inputted from a video capture device (e.g., a camera), a video archive (e.g., a video file stored in a storage device) containing previously captured video, or a video feed interface (e.g., a video broadcast transceiver) to receive video from a video content provider.

[039] As shown in **FIG. 1**, video sequence 100 can include a series of pictures arranged temporally along a timeline, including pictures 102, 104, 106, and 108. Pictures 102-106 are continuous, and there are more pictures between pictures 106 and 108. In **FIG. 1**, picture 102 is an I-picture, the reference picture of which is picture 102 itself. Picture 104 is a P-picture, the reference picture of which is picture 102, as indicated by the arrow. Picture 106 is a B-picture, the reference pictures of which are pictures 104 and 108, as indicated by the arrows. In some embodiments, the reference picture of a picture (e.g., picture 104) can be not immediately preceding or following the picture. For example, the reference picture of picture 104 can be a picture preceding picture 102. It should be noted that the reference

pictures of pictures 102-106 are only examples, and this disclosure does not limit embodiments of the reference pictures as the examples shown in **FIG. 1**.

[040] Typically, video codecs do not encode or decode an entire picture at one time due to the computing complexity of such tasks. Rather, they can split the picture into basic segments, and encode or decode the picture segment by segment. Such basic segments are referred to as basic processing units ("BPUs") in this disclosure. For example, structure 110 in **FIG. 1** shows an example structure of a picture of video sequence 100 (e.g., any of pictures 102-108). In structure 110, a picture is divided into 4×4 basic processing units, the boundaries of which are shown as dash lines. In some embodiments, the basic processing units can be referred to as "macroblocks" in some video coding standards (e.g., MPEG family, H.261, H.263, or H.264/AVC), or as "coding tree units" ("CTUs") in some other video coding standards (e.g., H.265/HEVC or H.266/VVC). The basic processing units can have variable sizes in a picture, such as 128×128, 64×64, 32×32, 16×16, 4×8, 16×32, or any arbitrary shape and size of pixels. The sizes and shapes of the basic processing units can be selected for a picture based on the balance of coding efficiency and levels of details to be kept in the basic processing unit.

[041] The basic processing units can be logical units, which can include a group of different types of video data stored in a computer memory (e.g., in a video frame buffer). For example, a basic processing unit of a color picture can include a luma component (Y) representing achromatic brightness information, one or more chroma components (e.g., Cb and Cr) representing color information, and associated syntax elements, in which the luma and chroma components can have the same size of the basic processing unit. The luma and chroma components can be referred to as "coding tree blocks" ("CTBs") in some video coding standards (e.g., H.265/HEVC or H.266/VVC). Any operation performed to a basic processing unit can be repeatedly performed to each of its luma and chroma components.

[042] Video coding has multiple stages of operations, examples of which will be detailed in **FIGs. 2A-2B** and **3A-3B**. For each stage, the size of the basic processing units can still be too large for processing, and thus can be further divided into segments referred to as "basic processing sub-units" in this disclosure. In some embodiments, the basic processing sub-units can be referred to as "blocks" in some video coding standards (e.g., MPEG family, H.261, H.263, or H.264/AVC), or as "coding units" ("CUs") in some other video coding standards (e.g., H.265/HEVC or H.266/VVC). A basic processing sub-unit can have the same or smaller size than the basic processing unit. Similar to the basic processing units, basic processing sub-units are also logical units, which can include a group of different types of video data (e.g., Y, Cb, Cr, and associated syntax elements) stored in a computer memory (e.g., in a video frame buffer). Any operation performed to a basic processing sub-unit can be repeatedly performed to each of its luma and chroma components. It should be noted that such division can be performed to further levels depending on processing needs. It should also be noted that different stages can divide the basic processing units using different schemes.

[043] For example, at a mode decision stage (an example of which will be detailed in **FIG. 2B**), the encoder can decide what prediction mode (e.g., intra-picture prediction or inter-picture prediction) to use for a basic processing unit, which can be too large to make such a decision. The encoder can split the basic processing unit into multiple basic processing sub-units (e.g., CUs as in H.265/HEVC or H.266/VVC), and decide a prediction type for each individual basic processing sub-unit.

[044] For another example, at a prediction stage (an example of which will be detailed in **FIG. 2A**), the encoder can perform prediction operation at the level of basic processing sub-units (e.g., CUs). However, in some cases, a basic processing sub-unit can still be too large to process. The encoder can further split the basic processing sub-unit into

smaller segments (e.g., referred to as "prediction blocks" or "PBs" in H.265/HEVC or H.266/VVC), at the level of which the prediction operation can be performed.

[045] For another example, at a transform stage (an example of which will be detailed in **FIG. 2A**), the encoder can perform a transform operation for residual basic processing sub-units (e.g., CUs). However, in some cases, a basic processing sub-unit can still be too large to process. The encoder can further split the basic processing sub-unit into smaller segments (e.g., referred to as "transform blocks" or "TBs" in H.265/HEVC or H.266/VVC), at the level of which the transform operation can be performed. It should be noted that the division schemes of the same basic processing sub-unit can be different at the prediction stage and the transform stage. For example, in H.265/HEVC or H.266/VVC, the prediction blocks and transform blocks of the same CU can have different sizes and numbers.

[046] In structure 110 of **FIG. 1**, basic processing unit 112 is further divided into 3×3 basic processing sub-units, the boundaries of which are shown as dotted lines. Different basic processing units of the same picture can be divided into basic processing sub-units in different schemes.

[047] In some implementations, to provide the capability of parallel processing and error resilience to video encoding and decoding, a picture can be divided into regions for processing, such that, for a region of the picture, the encoding or decoding process can depend on no information from any other region of the picture. In other words, each region of the picture can be processed independently. By doing so, the codec can process different regions of a picture in parallel, thus increasing the coding efficiency. Also, when data of a region is corrupted in the processing or lost in network transmission, the codec can correctly encode or decode other regions of the same picture without reliance on the corrupted or lost data, thus providing the capability of error resilience. In some video coding standards, a picture can be divided into different types of regions. For example, H.265/HEVC and

H.266/VVC provide two types of regions: "slices" and "tiles." It should also be noted that different pictures of video sequence 100 can have different partition schemes for dividing a picture into regions.

[048] For example, in **FIG. 1**, structure 110 is divided into three regions 114, 116, and 118, the boundaries of which are shown as solid lines inside structure 110. Region 114 includes four basic processing units. Each of regions 116 and 118 includes six basic processing units. It should be noted that the basic processing units, basic processing sub-units, and regions of structure 110 in **FIG. 1** are only examples, and this disclosure does not limit embodiments thereof.

[049] **FIG. 2A** illustrates a schematic diagram of an example encoding process 200A, consistent with embodiments of the disclosure. For example, the encoding process 200A can be performed by an encoder. As shown in **FIG. 2A**, the encoder can encode video sequence 202 into video bitstream 228 according to process 200A. Similar to video sequence 100 in **FIG. 1**, video sequence 202 can include a set of pictures (referred to as "original pictures") arranged in a temporal order. Similar to structure 110 in **FIG. 1**, each original picture of video sequence 202 can be divided by the encoder into basic processing units, basic processing sub-units, or regions for processing. In some embodiments, the encoder can perform process 200A at the level of basic processing units for each original picture of video sequence 202. For example, the encoder can perform process 200A in an iterative manner, in which the encoder can encode a basic processing unit in one iteration of process 200A. In some embodiments, the encoder can perform process 200A in parallel for regions (e.g., regions 114-118) of each original picture of video sequence 202.

[050] In **FIG. 2A**, the encoder can feed a basic processing unit (referred to as an "original BPU") of an original picture of video sequence 202 to prediction stage 204 to generate prediction data 206 and predicted BPU 208. The encoder can subtract predicted

BPU 208 from the original BPU to generate residual BPU 210. The encoder can feed residual BPU 210 to transform stage 212 and quantization stage 214 to generate quantized transform coefficients 216. The encoder can feed prediction data 206 and quantized transform coefficients 216 to binary coding stage 226 to generate video bitstream 228. Components 202, 204, 206, 208, 210, 212, 214, 216, 226, and 228 can be referred to as a "forward path." During process 200A, after quantization stage 214, the encoder can feed quantized transform coefficients 216 to inverse quantization stage 218 and inverse transform stage 220 to generate reconstructed residual BPU 222. The encoder can add reconstructed residual BPU 222 to predicted BPU 208 to generate prediction reference 224, which is used in prediction stage 204 for the next iteration of process 200A. Components 218, 220, 222, and 224 of process 200A can be referred to as a "reconstruction path." The reconstruction path can be used to ensure that both the encoder and the decoder use the same reference data for prediction.

[051] The encoder can perform process 200A iteratively to encode each original BPU of the original picture (in the forward path) and generate predicted reference 224 for encoding the next original BPU of the original picture (in the reconstruction path). After encoding all original BPUs of the original picture, the encoder can proceed to encode the next picture in video sequence 202.

[052] Referring to process 200A, the encoder can receive video sequence 202 generated by a video capturing device (e.g., a camera). The term "receive" used herein can refer to receiving, inputting, acquiring, retrieving, obtaining, reading, accessing, or any action in any manner for inputting data.

[053] At prediction stage 204, at a current iteration, the encoder can receive an original BPU and prediction reference 224, and perform a prediction operation to generate prediction data 206 and predicted BPU 208. Prediction reference 224 can be generated from the reconstruction path of the previous iteration of process 200A. The purpose of prediction

stage 204 is to reduce information redundancy by extracting prediction data 206 that can be used to reconstruct the original BPU as predicted BPU 208 from prediction data 206 and prediction reference 224.

[054] Ideally, predicted BPU 208 can be identical to the original BPU. However, due to non-ideal prediction and reconstruction operations, predicted BPU 208 is generally slightly different from the original BPU. For recording such differences, after generating predicted BPU 208, the encoder can subtract it from the original BPU to generate residual BPU 210. For example, the encoder can subtract values (e.g., greyscale values or RGB values) of pixels of predicted BPU 208 from values of corresponding pixels of the original BPU. Each pixel of residual BPU 210 can have a residual value as a result of such subtraction between the corresponding pixels of the original BPU and predicted BPU 208. Compared with the original BPU, prediction data 206 and residual BPU 210 can have fewer bits, but they can be used to reconstruct the original BPU without significant quality deterioration. Thus, the original BPU is compressed.

[055] To further compress residual BPU 210, at transform stage 212, the encoder can reduce spatial redundancy of residual BPU 210 by decomposing it into a set of two-dimensional "base patterns," each base pattern being associated with a "transform coefficient." The base patterns can have the same size (e.g., the size of residual BPU 210). Each base pattern can represent a variation frequency (e.g., frequency of brightness variation) component of residual BPU 210. None of the base patterns can be reproduced from any combinations (e.g., linear combinations) of any other base patterns. In other words, the decomposition can decompose variations of residual BPU 210 into a frequency domain. Such a decomposition is analogous to a discrete Fourier transform of a function, in which the base patterns are analogous to the base functions (e.g., trigonometry functions) of the discrete

Fourier transform, and the transform coefficients are analogous to the coefficients associated with the base functions.

[056] Different transform algorithms can use different base patterns. Various transform algorithms can be used at transform stage 212, such as, for example, a discrete cosine transform, a discrete sine transform, or the like. The transform at transform stage 212 is invertible. That is, the encoder can restore residual BPU 210 by an inverse operation of the transform (referred to as an "inverse transform"). For example, to restore a pixel of residual BPU 210, the inverse transform can be multiplying values of corresponding pixels of the base patterns by respective associated coefficients and adding the products to produce a weighted sum. For a video coding standard, both the encoder and decoder can use the same transform algorithm (thus the same base patterns). Thus, the encoder can record only the transform coefficients, from which the decoder can reconstruct residual BPU 210 without receiving the base patterns from the encoder. Compared with residual BPU 210, the transform coefficients can have fewer bits, but they can be used to reconstruct residual BPU 210 without significant quality deterioration. Thus, residual BPU 210 is further compressed.

[057] The encoder can further compress the transform coefficients at quantization stage 214. In the transform process, different base patterns can represent different variation frequencies (e.g., brightness variation frequencies). Because human eyes are generally better at recognizing low-frequency variation, the encoder can disregard information of high-frequency variation without causing significant quality deterioration in decoding. For example, at quantization stage 214, the encoder can generate quantized transform coefficients 216 by dividing each transform coefficient by an integer value (referred to as a "quantization parameter") and rounding the quotient to its nearest integer. After such an operation, some transform coefficients of the high-frequency base patterns can be converted to zero, and the transform coefficients of the low-frequency base patterns can be converted to smaller

integers. The encoder can disregard the zero-value quantized transform coefficients 216, by which the transform coefficients are further compressed. The quantization process is also invertible, in which quantized transform coefficients 216 can be reconstructed to the transform coefficients in an inverse operation of the quantization (referred to as "inverse quantization").

[058] Because the encoder disregards the remainders of such divisions in the rounding operation, quantization stage 214 can be lossy. Typically, quantization stage 214 can contribute the most information loss in process 200A. The larger the information loss is, the fewer bits the quantized transform coefficients 216 can need. For obtaining different levels of information loss, the encoder can use different values of the quantization parameter or any other parameter of the quantization process.

[059] At binary coding stage 226, the encoder can encode prediction data 206 and quantized transform coefficients 216 using a binary coding technique, such as, for example, entropy coding, variable length coding, arithmetic coding, Huffman coding, context-adaptive binary arithmetic coding, or any other lossless or lossy compression algorithm. In some embodiments, besides prediction data 206 and quantized transform coefficients 216, the encoder can encode other information at binary coding stage 226, such as, for example, a prediction mode used at prediction stage 204, parameters of the prediction operation, a transform type at transform stage 212, parameters of the quantization process (e.g., quantization parameters), an encoder control parameter (e.g., a bitrate control parameter), or the like. The encoder can use the output data of binary coding stage 226 to generate video bitstream 228. In some embodiments, video bitstream 228 can be further packetized for network transmission.

[060] Referring to the reconstruction path of process 200A, at inverse quantization stage 218, the encoder can perform inverse quantization on quantized transform coefficients

216 to generate reconstructed transform coefficients. At inverse transform stage 220, the encoder can generate reconstructed residual BPU 222 based on the reconstructed transform coefficients. The encoder can add reconstructed residual BPU 222 to predicted BPU 208 to generate prediction reference 224 that is to be used in the next iteration of process 200A.

[061] It should be noted that other variations of the process 200A can be used to encode video sequence 202. In some embodiments, stages of process 200A can be performed by the encoder in different orders. In some embodiments, one or more stages of process 200A can be combined into a single stage. In some embodiments, a single stage of process 200A can be divided into multiple stages. For example, transform stage 212 and quantization stage 214 can be combined into a single stage. In some embodiments, process 200A can include additional stages. In some embodiments, process 200A can omit one or more stages in **FIG. 2A**.

[062] **FIG. 2B** illustrates a schematic diagram of another example encoding process 200B, consistent with embodiments of the disclosure. Process 200B can be modified from process 200A. For example, process 200B can be used by an encoder conforming to a hybrid video coding standard (e.g., H.26x series). Compared with process 200A, the forward path of process 200B additionally includes mode decision stage 230 and divides prediction stage 204 into spatial prediction stage 2042 and temporal prediction stage 2044. The reconstruction path of process 200B additionally includes loop filter stage 232 and buffer 234.

[063] Generally, prediction techniques can be categorized into two types: spatial prediction and temporal prediction. Spatial prediction (e.g., an intra-picture prediction or "intra prediction") can use pixels from one or more already coded neighboring BPUs in the same picture to predict the current BPU. That is, prediction reference 224 in the spatial prediction can include the neighboring BPUs. The spatial prediction can reduce the inherent spatial redundancy of the picture. Temporal prediction (e.g., an inter-picture prediction or

"inter prediction") can use regions from one or more already coded pictures to predict the current BPU. That is, prediction reference 224 in the temporal prediction can include the coded pictures. The temporal prediction can reduce the inherent temporal redundancy of the pictures.

[064] Referring to process 200B, in the forward path, the encoder performs the prediction operation at spatial prediction stage 2042 and temporal prediction stage 2044. For example, at spatial prediction stage 2042, the encoder can perform the intra prediction. For an original BPU of a picture being encoded, prediction reference 224 can include one or more neighboring BPUs that have been encoded (in the forward path) and reconstructed (in the reconstructed path) in the same picture. The encoder can generate predicted BPU 208 by extrapolating the neighboring BPUs. The extrapolation technique can include, for example, a linear extrapolation or interpolation, a polynomial extrapolation or interpolation, or the like. In some embodiments, the encoder can perform the extrapolation at the pixel level, such as by extrapolating values of corresponding pixels for each pixel of predicted BPU 208. The neighboring BPUs used for extrapolation can be located with respect to the original BPU from various directions, such as in a vertical direction (e.g., on top of the original BPU), a horizontal direction (e.g., to the left of the original BPU), a diagonal direction (e.g., to the down-left, down-right, up-left, or up-right of the original BPU), or any direction defined in the used video coding standard. For the intra prediction, prediction data 206 can include, for example, locations (e.g., coordinates) of the used neighboring BPUs, sizes of the used neighboring BPUs, parameters of the extrapolation, a direction of the used neighboring BPUs with respect to the original BPU, or the like.

[065] For another example, at temporal prediction stage 2044, the encoder can perform the inter prediction. For an original BPU of a current picture, prediction reference 224 can include one or more pictures (referred to as "reference pictures") that have been

encoded (in the forward path) and reconstructed (in the reconstructed path). In some embodiments, a reference picture can be encoded and reconstructed BPU by BPU. For example, the encoder can add reconstructed residual BPU 222 to predicted BPU 208 to generate a reconstructed BPU. When all reconstructed BPUs of the same picture are generated, the encoder can generate a reconstructed picture as a reference picture. The encoder can perform an operation of "motion estimation" to search for a matching region in a scope (referred to as a "search window") of the reference picture. The location of the search window in the reference picture can be determined based on the location of the original BPU in the current picture. For example, the search window can be centered at a location having the same coordinates in the reference picture as the original BPU in the current picture and can be extended out for a predetermined distance. When the encoder identifies (e.g., by using a pel-recursive algorithm, a block-matching algorithm, or the like) a region similar to the original BPU in the search window, the encoder can determine such a region as the matching region. The matching region can have different dimensions (e.g., being smaller than, equal to, larger than, or in a different shape) from the original BPU. Because the reference picture and the current picture are temporally separated in the timeline (e.g., as shown in **FIG. 1**), it can be deemed that the matching region "moves" to the location of the original BPU as time goes by. The encoder can record the direction and distance of such a motion as a "motion vector." When multiple reference pictures are used (e.g., as picture 106 in **FIG. 1**), the encoder can search for a matching region and determine its associated motion vector for each reference picture. In some embodiments, the encoder can assign weights to pixel values of the matching regions of respective matching reference pictures.

[066] The motion estimation can be used to identify various types of motions, such as, for example, translations, rotations, zooming, or the like. For inter prediction, prediction data 206 can include, for example, locations (e.g., coordinates) of the matching region, the

motion vectors associated with the matching region, the number of reference pictures, weights associated with the reference pictures, or the like.

[067] For generating predicted BPU 208, the encoder can perform an operation of "motion compensation." The motion compensation can be used to reconstruct predicted BPU 208 based on prediction data 206 (e.g., the motion vector) and prediction reference 224. For example, the encoder can move the matching region of the reference picture according to the motion vector, in which the encoder can predict the original BPU of the current picture. When multiple reference pictures are used (e.g., as picture 106 in **FIG. 1**), the encoder can move the matching regions of the reference pictures according to the respective motion vectors and average pixel values of the matching regions. In some embodiments, if the encoder has assigned weights to pixel values of the matching regions of respective matching reference pictures, the encoder can add a weighted sum of the pixel values of the moved matching regions.

[068] In some embodiments, the inter prediction can be unidirectional or bidirectional. Unidirectional inter predictions can use one or more reference pictures in the same temporal direction with respect to the current picture. For example, picture 104 in **FIG. 1** is a unidirectional inter-predicted picture, in which the reference picture (i.e., picture 102) precedes picture 104. Bidirectional inter predictions can use one or more reference pictures at both temporal directions with respect to the current picture. For example, picture 106 in **FIG. 1** is a bidirectional inter-predicted picture, in which the reference pictures (i.e., pictures 104 and 108) are at both temporal directions with respect to picture 104.

[069] Still referring to the forward path of process 200B, after spatial prediction 2042 and temporal prediction stage 2044, at mode decision stage 230, the encoder can select a prediction mode (e.g., one of the intra prediction or the inter prediction) for the current iteration of process 200B. For example, the encoder can perform a rate-distortion

optimization technique, in which the encoder can select a prediction mode to minimize a value of a cost function depending on a bit rate of a candidate prediction mode and distortion of the reconstructed reference picture under the candidate prediction mode. Depending on the selected prediction mode, the encoder can generate the corresponding predicted BPU 208 and predicted data 206.

[070] In the reconstruction path of process 200B, if intra prediction mode has been selected in the forward path, after generating prediction reference 224 (e.g., the current BPU that has been encoded and reconstructed in the current picture), the encoder can directly feed prediction reference 224 to spatial prediction stage 2042 for later usage (e.g., for extrapolation of a next BPU of the current picture). If the inter prediction mode has been selected in the forward path, after generating prediction reference 224 (e.g., the current picture in which all BPUs have been encoded and reconstructed), the encoder can feed prediction reference 224 to loop filter stage 232, at which the encoder can apply a loop filter to prediction reference 224 to reduce or eliminate distortion (e.g., blocking artifacts) introduced by the inter prediction. The encoder can apply various loop filter techniques at loop filter stage 232, such as, for example, deblocking, sample adaptive offsets, adaptive loop filters, or the like. The loop-filtered reference picture can be stored in buffer 234 (or "decoded picture buffer") for later use (e.g., to be used as an inter-prediction reference picture for a future picture of video sequence 202). The encoder can store one or more reference pictures in buffer 234 to be used at temporal prediction stage 2044. In some embodiments, the encoder can encode parameters of the loop filter (e.g., a loop filter strength) at binary coding stage 226, along with quantized transform coefficients 216, prediction data 206, and other information.

[071] **FIG. 3A** illustrates a schematic diagram of an example decoding process 300A, consistent with embodiments of the disclosure. Process 300A can be a decompression

process corresponding to the compression process 200A in **FIG. 2A**. In some embodiments, process 300A can be similar to the reconstruction path of process 200A. A decoder can decode video bitstream 228 into video stream 304 according to process 300A. Video stream 304 can be very similar to video sequence 202. However, due to the information loss in the compression and decompression process (e.g., quantization stage 214 in **FIGs. 2A-2B**), generally, video stream 304 is not identical to video sequence 202. Similar to processes 200A and 200B in **FIGs. 2A-2B**, the decoder can perform process 300A at the level of basic processing units (BPUs) for each picture encoded in video bitstream 228. For example, the decoder can perform process 300A in an iterative manner, in which the decoder can decode a basic processing unit in one iteration of process 300A. In some embodiments, the decoder can perform process 300A in parallel for regions (e.g., regions 114-118) of each picture encoded in video bitstream 228.

[072] In **FIG. 3A**, the decoder can feed a portion of video bitstream 228 associated with a basic processing unit (referred to as an "encoded BPU") of an encoded picture to binary decoding stage 302. At binary decoding stage 302, the decoder can decode the portion into prediction data 206 and quantized transform coefficients 216. The decoder can feed quantized transform coefficients 216 to inverse quantization stage 218 and inverse transform stage 220 to generate reconstructed residual BPU 222. The decoder can feed prediction data 206 to prediction stage 204 to generate predicted BPU 208. The decoder can add reconstructed residual BPU 222 to predicted BPU 208 to generate predicted reference 224. In some embodiments, predicted reference 224 can be stored in a buffer (e.g., a decoded picture buffer in a computer memory). The decoder can feed predicted reference 224 to prediction stage 204 for performing a prediction operation in the next iteration of process 300A.

[073] The decoder can perform process 300A iteratively to decode each encoded BPU of the encoded picture and generate predicted reference 224 for encoding the next

encoded BPU of the encoded picture. After decoding all encoded BPUs of the encoded picture, the decoder can output the picture to video stream 304 for display and proceed to decode the next encoded picture in video bitstream 228.

[074] At binary decoding stage 302, the decoder can perform an inverse operation of the binary coding technique used by the encoder (e.g., entropy coding, variable length coding, arithmetic coding, Huffman coding, context-adaptive binary arithmetic coding, or any other lossless compression algorithm). In some embodiments, besides prediction data 206 and quantized transform coefficients 216, the decoder can decode other information at binary decoding stage 302, such as, for example, a prediction mode, parameters of the prediction operation, a transform type, parameters of the quantization process (e.g., quantization parameters), an encoder control parameter (e.g., a bitrate control parameter), or the like. In some embodiments, if video bitstream 228 is transmitted over a network in packets, the decoder can depacketize video bitstream 228 before feeding it to binary decoding stage 302.

[075] **FIG. 3B** illustrates a schematic diagram of another example decoding process 300B, consistent with embodiments of the disclosure. Process 300B can be modified from process 300A. For example, process 300B can be used by a decoder conforming to a hybrid video coding standard (e.g., H.26x series). Compared with process 300A, process 300B additionally divides prediction stage 204 into spatial prediction stage 2042 and temporal prediction stage 2044, and additionally includes loop filter stage 232 and buffer 234.

[076] In process 300B, for an encoded basic processing unit (referred to as a "current BPU") of an encoded picture (referred to as a "current picture") that is being decoded, prediction data 206 decoded from binary decoding stage 302 by the decoder can include various types of data, depending on what prediction mode was used to encode the current BPU by the encoder. For example, if intra prediction was used by the encoder to encode the current BPU, prediction data 206 can include a prediction mode indicator (e.g., a

flag value) indicative of the intra prediction, parameters of the intra prediction operation, or the like. The parameters of the intra prediction operation can include, for example, locations (e.g., coordinates) of one or more neighboring BPUs used as a reference, sizes of the neighboring BPUs, parameters of extrapolation, a direction of the neighboring BPUs with respect to the original BPU, or the like. For another example, if inter prediction was used by the encoder to encode the current BPU, prediction data 206 can include a prediction mode indicator (e.g., a flag value) indicative of the inter prediction, parameters of the inter prediction operation, or the like. The parameters of the inter prediction operation can include, for example, the number of reference pictures associated with the current BPU, weights respectively associated with the reference pictures, locations (e.g., coordinates) of one or more matching regions in the respective reference pictures, one or more motion vectors respectively associated with the matching regions, or the like.

[077] Based on the prediction mode indicator, the decoder can decide whether to perform a spatial prediction (e.g., the intra prediction) at spatial prediction stage 2042 or a temporal prediction (e.g., the inter prediction) at temporal prediction stage 2044. The details of performing such spatial prediction or temporal prediction are described in **FIG. 2B** and will not be repeated hereinafter. After performing such spatial prediction or temporal prediction, the decoder can generate predicted BPU 208. The decoder can add predicted BPU 208 and reconstructed residual BPU 222 to generate prediction reference 224, as described in **FIG. 3A**.

[078] In process 300B, the decoder can feed predicted reference 224 to spatial prediction stage 2042 or temporal prediction stage 2044 for performing a prediction operation in the next iteration of process 300B. For example, if the current BPU is decoded using the intra prediction at spatial prediction stage 2042, after generating prediction reference 224 (e.g., the decoded current BPU), the decoder can directly feed prediction reference 224 to

spatial prediction stage 2042 for later usage (e.g., for extrapolation of a next BPU of the current picture). If the current BPU is decoded using the inter prediction at temporal prediction stage 2044, after generating prediction reference 224 (e.g., a reference picture in which all BPUs have been decoded), the encoder can feed prediction reference 224 to loop filter stage 232 to reduce or eliminate distortion (e.g., blocking artifacts). The decoder can apply a loop filter to prediction reference 224, in a way as described in **FIG. 2B**. The loop-filtered reference picture can be stored in buffer 234 (e.g., a decoded picture buffer in a computer memory) for later use (e.g., to be used as an inter-prediction reference picture for a future encoded picture of video bitstream 228). The decoder can store one or more reference pictures in buffer 234 to be used at temporal prediction stage 2044. In some embodiments, when the prediction mode indicator of prediction data 206 indicates that inter prediction was used to encode the current BPU, prediction data can further include parameters of the loop filter (e.g., a loop filter strength).

[079] **FIG. 4** is a block diagram of an example apparatus 400 for encoding or decoding a video, consistent with embodiments of the disclosure. As shown in **FIG. 4**, apparatus 400 can include processor 402. When processor 402 executes instructions described herein, apparatus 400 can become a specialized machine for video encoding or decoding. Processor 402 can be any type of circuitry capable of manipulating or processing information. For example, processor 402 can include any combination of any number of a central processing unit (or "CPU"), a graphics processing unit (or "GPU"), a neural processing unit ("NPU"), a microcontroller unit ("MCU"), an optical processor, a programmable logic controller, a microcontroller, a microprocessor, a digital signal processor, an intellectual property (IP) core, a Programmable Logic Array (PLA), a Programmable Array Logic (PAL), a Generic Array Logic (GAL), a Complex Programmable Logic Device (CPLD), a Field-Programmable Gate Array (FPGA), a System On Chip (SoC),

an Application-Specific Integrated Circuit (ASIC), or the like. In some embodiments, processor 402 can also be a set of processors grouped as a single logical component. For example, as shown in **FIG. 4**, processor 402 can include multiple processors, including processor 402a, processor 402b, and processor 402n.

[080] Apparatus 400 can also include memory 404 configured to store data (e.g., a set of instructions, computer codes, intermediate data, or the like). For example, as shown in **FIG. 4**, the stored data can include program instructions (e.g., program instructions for implementing the stages in processes 200A, 200B, 300A, or 300B) and data for processing (e.g., video sequence 202, video bitstream 228, or video stream 304). Processor 402 can access the program instructions and data for processing (e.g., via bus 410), and execute the program instructions to perform an operation or manipulation on the data for processing. Memory 404 can include a high-speed random-access storage device or a non-volatile storage device. In some embodiments, memory 404 can include any combination of any number of a random-access memory (RAM), a read-only memory (ROM), an optical disc, a magnetic disk, a hard drive, a solid-state drive, a flash drive, a security digital (SD) card, a memory stick, a compact flash (CF) card, or the like. Memory 404 can also be a group of memories (not shown in **FIG. 4**) grouped as a single logical component.

[081] Bus 410 can be a communication device that transfers data between components inside apparatus 400, such as an internal bus (e.g., a CPU-memory bus), an external bus (e.g., a universal serial bus port, a peripheral component interconnect express port), or the like.

[082] For ease of explanation without causing ambiguity, processor 402 and other data processing circuits are collectively referred to as a "data processing circuit" in this disclosure. The data processing circuit can be implemented entirely as hardware, or as a combination of software, hardware, or firmware. In addition, the data processing circuit can

be a single independent module or can be combined entirely or partially into any other component of apparatus 400.

[083] Apparatus 400 can further include network interface 406 to provide wired or wireless communication with a network (e.g., the Internet, an intranet, a local area network, a mobile communications network, or the like). In some embodiments, network interface 406 can include any combination of any number of a network interface controller (NIC), a radio frequency (RF) module, a transponder, a transceiver, a modem, a router, a gateway, a wired network adapter, a wireless network adapter, a Bluetooth adapter, an infrared adapter, an near-field communication ("NFC") adapter, a cellular network chip, or the like.

[084] In some embodiments, optionally, apparatus 400 can further include peripheral interface 408 to provide a connection to one or more peripheral devices. As shown in **FIG. 4**, the peripheral device can include, but is not limited to, a cursor control device (e.g., a mouse, a touchpad, or a touchscreen), a keyboard, a display (e.g., a cathode-ray tube display, a liquid crystal display, or a light-emitting diode display), a video input device (e.g., a camera or an input interface coupled to a video archive), or the like.

[085] It should be noted that video codecs (e.g., a codec performing process 200A, 200B, 300A, or 300B) can be implemented as any combination of any software or hardware modules in apparatus 400. For example, some or all stages of process 200A, 200B, 300A, or 300B can be implemented as one or more software modules of apparatus 400, such as program instructions that can be loaded into memory 404. For another example, some or all stages of process 200A, 200B, 300A, or 300B can be implemented as one or more hardware modules of apparatus 400, such as a specialized data processing circuit (e.g., an FPGA, an ASIC, an NPU, or the like).

[086] One of the key requirements of the VVC standard is to offer video conferencing applications the ability to tolerate diversity of networks and devices, and to be

able to rapidly adapt to varying network environments, including rapidly reducing encoded bit rate when network conditions deteriorate, and to rapidly increasing video quality when network conditions improve. The expected video quality may vary from very low to very high. The standard shall also 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). During switching from one representation to another representation (such as switching from one resolution to another resolution), the standard shall enable the use of efficient prediction structure without compromising the fast and seamless switching capability.

[087] The aim of adaptive resolution change (ARC) is to allow a stream to change spatial resolution between coded pictures within the same video sequence, with or without requiring a new IDR frame and with or without requiring multi-layers as in scalable video codec. Instead, at a switch point, pictures change resolution and may be predicted from reference pictures with the same resolution (if available) and from reference pictures of a different resolution. If a reference picture is of a different resolution, then the reference picture is resampled as shown in **FIG. 5.** For example, as shown in **FIG. 5**, a resolution of a reference picture 506 is the same as a current picture 502, but resolutions of reference pictures 504 and 508 are different from that of the current picture 502. After reference pictures 504 and 508 are resampled to the resolution of the current picture 502, motion compensated prediction from these references may be performed. Hence, adaptive resolution change (ARC) is also sometimes referred to as reference picture resampling (RPR), and these two terms are used interchangeably in this disclosure.

[088] When the resolution of a reference frame is different from that of the current frame, one way to generate the motion compensated prediction signal is picture-based resampling, where the reference picture is first resampled to the same resolution as the

current picture, and the existing motion compensation process with motion vectors can be applied. The motion vectors may be scaled (if they are sent in units before resampling is applied), or not scaled (if they are sent in units after resampling is applied). With the picture-based resampling, in particular for reference picture down-sampling (that is, resolution of the reference picture is larger than that of the current picture), information may be lost in the reference resampling step before the motion compensated interpolation, because down-sampling is usually achieved with a low-pass filtering followed by decimation).

[089] Another way is block-based resampling, where resampling is performed at the block level. This is done by examining the reference picture(s) used by the current block, and if one or both of them have different resolutions than the current picture, than resampling is performed in combination with the sub-pel motion compensated interpolation process.

[090] In the block-based resampling, combining the resampling and motion compensated interpolation into one filtering operation may reduce the information loss mentioned above. Take the following case as an example: the motion vector of the current block has half-pel precision in one dimension (e.g., the horizontal dimension), and the reference picture's width is 2 times that of the current picture. In this case, compared to the picture-level resampling, which will reduce the width of the reference picture by half to match the width of the current picture, and then doing half-pel motion interpolation, the block-based resampling method can directly fetch the odd positions in the reference pictures as the reference block at half-pel precision. In the 15th JVET meeting, a block-based resampling method for ARC was adopted in VVC, where the motion compensation (MC) interpolation and reference resampling is combined and performed in one-step filter. In VVC draft 6, the existing filters for MC interpolation with no reference resampling are reused for the MC interpolation with reference resampling. The same filter is used for both reference up-sampling and reference down-sampling. The detail of filter selection is described below.

[091] For luma components, if half-pel AMVR mode is selected and interpolated position is half-pel, a 6-tap filter [3, 9, 20, 20, 9, 3] is used. If motion compensated block size is 4×4, the following 6-tap filters as shown in Table 6 of **FIG. 6** are used. And otherwise 8-tap filters as shown Table 7 of **FIG. 7** are used.

[092] For chroma components, the 4-tap filters shown in Table 8 of **FIG. 8** below are used.

[093] In VVC, the same filters are used for MC interpolation without reference resampling and MC interpolation with reference resampling. While the VVC motion compensated interpolation filter (MCIF) is designed based on DCT up-sampling, it may not be appropriate to use it as a one-step filter combining reference down-sampling and MC interpolation. For example, for phase 0 filtering (e.g., scaled motion vector is integer), the VVC 8-tap MCIF coefficients are [0, 0, 0, 64, 0, 0, 0, 0], which means the prediction sample is directly copied from the reference sample. While it may not be a problem for MC interpolation without reference down-sampling or with reference up-sampling, it may cause aliasing artifact for reference down-sampling case, due to the lack of low-pass filter before decimation.

[094] This present disclosure provides methods of using cosine windowed-sinc filters for MC interpolation with reference down-sampling.

[095] Windowed-sinc filters are band-pass filters that separate one band of frequency from others. The windowed-sinc filter is a low-pass filter with a frequency response that allows all frequencies below a cutoff frequency pass through with amplitude 1 and stops all frequencies above the cutoff frequency with zero amplitude, as shown in **FIG. 9**.

[096] The filter kernel, also known as the impulse response of the filter, is obtained by taking the Inverse Fourier Transform of the frequency response of the ideal low-pass

filter. The impulse response of the low-pass filter is in the general form of a sinc function based on Equation (1) below.

$$h(n) = \frac{fc}{r}\text{Sinc}\left(\frac{fc}{r}n\right), \quad n = -\infty, \ldots, +\infty \qquad \text{Equation (1)}$$

where fc is the cut-off frequency with the value in [0, 1], r is the down-sampling ratio, i.e., 1.5 for 1.5:1 down-sampling and 2 for the 2:1 down-sampling. The sinc function is defined based on Equation (2) below.

$$\text{Sinc}(x) = \begin{cases} 1, & x = 0 \\ \frac{\sin(\pi x)}{\pi x}, & x \neq 0 \end{cases} \qquad \text{Equation (2)}$$

[097] The sinc function is infinite. To make the filter kernel in finite length, a window function is applied to truncate the filter kernel to L points. To obtain a smooth tapered curve, a cosine-windowed function is used, which is based on Equation (3) below.

$$w(n) = \cos\left(\frac{\pi n}{L-1}\right), n = -\frac{L-1}{2}, \ldots, \frac{L-1}{2} \qquad \text{Equation (3)}$$

[098] The kernel of the cosine windowed-sinc filter is the product of the ideal response function $h(n)$ and the cosine-window function $w(n)$ based on Equation (4) below.

$$f(n) = h(n)w(n) = \frac{fc}{r}\text{Sinc}\left(\frac{fc}{r}n\right)\cos\left(\frac{\pi n}{L-1}\right), \quad n = -\frac{L-1}{2}, \ldots, \frac{L-1}{2} \qquad \text{Equation (4)}$$

[099] There are two parameters selected for a windowed-sinc kernel, the cutoff frequency fc and the kernel length L. By adjusting the value of L and fc, the desired filter response can be achieved. For instance, for down-sampling filters used in the scalable HEVC test model (SHM), $fc = 0.9$ and L=13.

[0100] Filter coefficients obtained in Equation (4) are real numbers. Applying a filter is equivalent to calculate the weighted average of reference samples, with the weights being the filter coefficients. For efficient calculation in digital computers or hardware, the coefficients are normalized, multiplied by a scaler and rounded to integers, such that the sum

of the coefficients is equal to 2^N, where N is an integer. The filtered sample is divided by 2^N (equivalent to right shift N bits). For example, in VVC draft 6, the sum of interpolation filter coefficients is 64.

[0101] In some embodiments, the down-sampling filters can be used in SHM for the VVC motion compensation interpolation with reference down-sampling, for both luma and chroma components, and the existing MCIF for motion compensation interpolation with reference up-sampling can be used. While the kernel length L=13, the first coefficient is small and rounded to zero, the filter length can be reduced to 12, without impact to the filter performance.

[0102] As an example, the filter coefficients for 2:1 down-sampling and 1.5:1 down-sampling are shown in Table 10 of **FIG. 10** and Table 11 of **FIG. 11**, respectively.

[0103] Besides the value of the coefficients, there are several other differences between the designs of SHM filter and the existing MCIF.

[0104] As a first difference, the SHM filter requires filtering in an integer sample position as well as in a fractional sample position, while the MCIF only requires filtering in the fractional sample position. An example of modification to the VVC draft 6 luma sample interpolation filtering process for reference down-sampling case is described in Table 12 of **FIG. 12** below.

[0105] An example of modification to the VVC draft 6 chroma sample interpolation filtering process for reference down-sampling case is described in Table 13 of **FIG. 13** below.

[0106] As a second difference, the sum of filter coefficients is 128 for SHM filters, while the sum of filter coefficients is 64 for the existing MCIF. In VVC draft 6, in order to reduce the loss caused by rounding error, the intermediate prediction signals are kept in a higher precision (represented by higher bit-depth) than the output signal. The precision of the

intermediate signal is called internal precision. In some embodiments, in order to keep the internal precision being the same as in VVC draft 6, the output of SHM filters needs to right shift 1 additional bit comparing to using existing MCIF. An example of modification to the VVC draft 6 luma sample interpolation filtering process for reference down-sampling case is shown in Table 14 of **FIG. 14** below.

[0107] An example of modification to the VVC draft 6 chroma sample interpolation filtering process for reference down-sampling case is shown in Table 15 of **FIG. 15** below.

[0108] According to some embodiments, the internal precision can be increased by 1-bit, and addition 1-bit right shift can be used to convert the internal precision to the output precision. An example of modification to the VVC draft 6 luma sample interpolation filtering process for reference down-sampling case is shown in Table 16 of **FIG. 16**.

[0109] An example of modification to the VVC draft 6 chroma sample interpolation filtering process for reference down-sampling case is shown in Table 17 of **FIG. 17** below.

[0110] As a third difference, SHM filter has 12 taps. Therefore, to generate an interpolated sample, 11 neighboring samples (5 to the left and 6 to the right, or 5 above or 6 below) are needed. Comparing to MCIF, additional neighboring samples are fetched. In VVC draft 6, the chroma mv precision is 1/32. However, the SHM filter has only 16 phases. Therefore, the chroma mv can be rounded to 1/16 for reference down-sampling. This can be done by right shifting the last 5 bit of the chroma mv by 1 bit. An example of modification to the VVC draft 6 chroma fractional sample position calculation for reference down-sampling case is shown in Table 18 of **FIG. 18**.

[0111] According to some embodiments, in order to align to the existing MCIF design in VVC draft, it is proposed to use 8-tap cosine windowed-sinc filters. The filter coefficients can be derived by setting L=9 in the cosine windowed-sinc function in Equation (4). The sum of filter coefficients may be set to 64 for further alignment to the existing MCIF

filters. In the disclosed embodiments, the filters for complementary phases may be symmetric (e.g., the filter coefficients of complementary phase are reversed) or non-symmetric. Example filter coefficients for 2:1 and 1.5:1 ratios are shown in Table 19 of **FIG. 19** and Table 20 of **FIG. 20**, respectively.

[0112] According to some embodiments, to adapt to the 1/32-sample precision for chroma component, a 32-phase cosine windowed-sinc filter set may be used in chroma motion compensation interpolation with reference down-sampling. Examples of filter coefficients for 2:1 and 1.5:1 ratios are shown in the following Table 21 of **FIG. 21** and Table 22 of **FIG. 22**, respectively.

[0113] According to some embodiments, for 4×4 luma block, 6-tap cosine windowed-sinc filter may be used for MC interpolation with reference down-sampling. Examples of filter coefficients for 2:1 and 1.5:1 ratios are shown in the following Table 23 of **FIG. 23** and Table 24 of **FIG. 24**, respectively.

[0114] In VVC, different interpolation filters are used in different motion compensation cases. For example, in regular motion compensation, 8-tap DCT-based interpolation filters are used. In motion compensation for 4×4 sub-blocks, 6-tap DCT-based interpolation filters are used. In motion compensation with MVD precision being 1/2 and MVD phase being 1/2, a different 6-tap filter is used. In chroma motion compensation, 4-tap DCT-based interpolation filters are used.

[0115] In some embodiments, the DCT-based interpolation filter can be replaced with the cosine windowed-sinc filter of the same filter length, when the reference picture has a higher resolution than the currently coded picture. In these embodiments, the 8-tap DCT-based interpolation filters can be replaced with 8-tap cosine windowed-sinc filter for regular motion compensation; the 6-tap DCT-based interpolation filters can be replaced with 6-tap cosine windowed-sinc filter for 4×4 sub-block motion compensation. The 4-tap DCT-based

interpolation filters can be replaced with 4-tap cosine-windowed-sinc filters for chroma motion compensation. Moreover, the 6-tap DCT-based interpolation filter may be used in motion compensation when MVD precision is ½ and phase is ½. And the filter selection may depend on ratio between the reference picture resolution and the current picture resolution. Exemplary 8-tap, 6-tap and 4-tap cosine-windowed-sinc filters for 2:1 and 1.5:1 down-sampling ratios are shown in Tables 19-24 **FIGs. 19-24**. The filters shown in Tables 25-30 of **FIGs. 25-30** are symmetric on complementary phases.

[0116] The filter coefficients obtained in Equation (4) are real values. After normalization, scaling, the filter coefficients can be rounded to integers. The sum of the filter coefficients, also known as filter gain, represents filter precision. In digital processing, the filter gain usually is set to 2^N. Due to the rounding operation, the sum of filter coefficients may not be equal to the filter gain (2^N). Additional adjustments to the filter coefficients are performed such that the sum of the filter coefficients is equal to the gain (2^N). In some embodiments, the adjustment includes a determination of a proper rounding direction. The rounding direction can include rounding up or rounding down. Therefore, the adjusted filter coefficients can be summed up to the filter gain to minimize/maximize a cost function. In some embodiments, the cost function may be the sum of absolute difference (SAD) or the sum of squared error (SSE) between the rounded filter coefficients and the real filter coefficients before rounding. In some embodiments, the cost function may be SAD or SSE between a frequency response of the rounded filter coefficients and a frequency response of a reference. The reference may be the real filter coefficients before rounding, or the frequency response of an ideal filter, or the frequency response of a filter with a longer length than the filter being designed. For example, in designing an 8-tap filter, the reference may be a 12-tap filter. Based on the above rounding methods, another set of cosine windowed sinc filters with 8-tap, 6-tap and 4-tap length is exemplified in Tables 31-36 of **FIGs. 31-36**.

[0117] Since in adaptive resolution change application, the source picture down-sampling and reconstructed picture up-sampling process may not be normatively defined in the standard, which means the user can use any re-sampling filters. While the user selected filter may not match the reference resampling filter, the performance may be degraded due to the filter mismatch. In some disclosed embodiments, the coefficients of reference resample filters are signaled in the bitstream. And the encoder/decoder may use the user defined filter for reference resampling. The signaling of filter coefficients is exemplified as in Tables 37 and 38 of **FIGs. 37-38**.

[0118] The syntax structures exemplified in Tables 37 and 38 may be presented in high-level syntax, such as Sequence Parameter Set (SPS), Picture Parameter Set (PPS), Adaptation Parameter Set (APS), slice header, and the like.

[0119] The use of the user-defined RPR filters and the presence of the above exemplary syntax structures may be controlled by flag(s) in the high-level syntax, including SPS, PPS, APS, slice header, and the like.

[0120] Consistent with the disclosed embodiments, the hardware implementation may be more efficient when one or more of the signaled filters use a fixed filter length and filter norm. Therefore, in some embodiments, the filter norm and/or filter taps may not be signaled and may instead be implied (e.g., implied to be default values defined in the standard). The filter norm, filter taps, filter precision, or maximum bit-depth may be part of the encoder conformance requirement.

[0121] In some embodiments, the encoder may send supplemental enhancement information (SEI) messages to suggest what non-normative up-sampling filters should be used at the receiving device after the pictures are decoded. Such SEI message is optional. However, with the SEI message, decoders that use the suggested up-sampling filters can

obtain better video quality. The filter coefficients() syntax structure in Table 31 (**FIG. 31**) can be used to specify the suggested up-sampling filters in such an SEI message.

[0122] **FIG. 39** is a flowchart of an exemplary method 3900 for processing video content, consistent with embodiments of the disclosure. In some embodiments, method 3900 can be performed by a codec (e.g., an encoder using encoding processes 200A or 200B in **FIGs. 2A-2B** or a decoder using decoding processes 300A or 300B in **FIGs. 3A-3B**). For example, the codec can be implemented as one or more software or hardware components of an apparatus (e.g., apparatus 400) for encoding or transcoding a video sequence. In some embodiments, the video sequence can be an uncompressed video sequence (e.g., video sequence 202) or a compressed video sequence that is decoded (e.g., video stream 304). In some embodiments, the video sequence can be a monitoring video sequence, which can be captured by a monitoring device (e.g., the video input device in **FIG. 4**) associated with a processor (e.g., processor 402) of the apparatus. The video sequence can include multiple pictures. The apparatus can perform method 3900 at the level of pictures. For example, the apparatus can process one picture at a time in method 3900. For another example, the apparatus can process a plurality of pictures at a time in method 3900. Method 3900 can include steps as below.

[0123] At step 3902, in response to a target picture and a reference picture having different resolutions, a band-pass filter can be applied to the reference picture, to perform a motion compensated interpolation with reference down-sampling to generate a reference block.

[0124] The band-pass filter can be a cosine windowed-sinc filter generated based on an ideal low-pass filter and a window function. For example, a kernel function $f(n)$ of the cosine windowed-sinc filter can be a product of the ideal low-pass filter $h(n)$ and the window function $w(n)$, which can be based on Equation (4).

[0125] Therefore, $f(n) = \frac{fc}{r}\text{Sinc}\left(\frac{fc}{r}n\right)\cos\left(\frac{\pi n}{L-1}\right),\ n = -\frac{L-1}{2}, \ldots, \frac{L-1}{2}$, wherein fc is a cutoff frequency of the cosine windowed-sinc filter $f(n)$, L being a kernel length, and r being a down-sampling ratio of the reference down-sampling.

[0126] Filter coefficients of the cosine windowed-sinc filter can be rounded before being applied. **FIG. 40** is a flowchart of an exemplary method 4000 for rounding filter coefficients of the cosine windowed-sinc filter, consistent with embodiments of the disclosure. It is appreciated that method 4000 can be implemented independently or as part of method 3900. Method 4000 can include steps as below.

[0127] At step 4002, real filter coefficients of the band-pass filter can be obtained.

[0128] At step 4004, a plurality of rounding directions for the real filter coefficients can be determined respectively.

[0129] At step 4006, a plurality of combinations of rounded filter coefficients can be generated by rounding the real filter coefficients according to the plurality of rounding directions, respectively.

[0130] At step 4008, among the plurality of combinations, a combination of rounded filter coefficients that causes a cost function to be minimum or maximum can be selected. In some embodiments, the cost function can be associated with the rounded filter coefficients and a reference. The reference can be the real filter coefficients of the band-pass filter before the rounding, a frequency response of an ideal filter, or a frequency response of a filter having a longer length than the band-pass filter. For example, the cost function can be a sum of absolute difference (SAD) or a sum of squared error (SSE) between the rounded filter coefficients and the real filter coefficients before rounding, when the reference can be the real filter coefficients of the band-pass filter before the rounding. As another example, the cost function can be SAD or SSE between a frequency response of the rounded filter coefficients and a frequency response of the ideal filter, when the reference is a frequency response of an

ideal filter. As discussed above, the reference can also be a frequency response of a filter having a longer length than the band-pass filter. For example, if the band-pass filter is an 8-tap filter, the reference can be a 12-tap filter, which has a longer length than the 8-tap band-pass filter.

[0131] The selected combination of rounded filter coefficients of the band-pass filter can be signaled in at least one of Sequence Parameter Set (SPS), a Picture Parameter Set (PPS), an Adaptation Parameter Set (APS), and a slice header, along with the syntax structures associated with the filter coefficients.

[0132] It is appreciated that the cosine windowed-sinc filter is an 8-tap filter, a 6-tap filter, or a 4-tap filter, as shown in Tables 19-36 (**FIGS. 19-36**). In some embodiments, the band-pass filter can be determined to be one of an 8-tap filter, a 6-tap filter, and a 4-tap filter, based on a ratio between the resolution of the reference picture and the resolution of the target picture.

[0133] In applying the band-pass filter to the reference picture, a luma sample or a chroma sample can be obtained at a fractional sample position. Filter coefficients for the obtained luma sample or chroma sample can be determined with reference to the look-up table (e.g., Tables 19-36) using the fractional sample position.

[0134] With reference back to **FIG. 39**, at step 3904, a block of the target picture can be processed using the reference block. For example, the block of the target picture can be coded or decoded using the reference block.

[0135] In some embodiments, a non-transitory computer-readable storage medium including instructions is also provided, and the instructions may be executed by a device (such as the disclosed encoder and decoder), for performing the above-described methods. Common forms of non-transitory media include, for example, a floppy disk, a flexible disk, hard disk, solid state drive, magnetic tape, or any other magnetic data storage medium, a CD-

ROM, any other optical data storage medium, any physical medium with patterns of holes, a RAM, a PROM, and EPROM, a FLASH-EPROM or any other flash memory, NVRAM, a cache, a register, any other memory chip or cartridge, and networked versions of the same. The device may include one or more processors (CPUs), an input/output interface, a network interface, and/or a memory.

[0136] The embodiments may further be described using the following clauses:

1. A computer-implemented method for performing motion compensation interpolation, comprising:

in response to a target picture and a reference picture having different resolutions, applying a band-pass filter to the reference picture, to perform a motion compensated interpolation with reference down-sampling to generate a reference block; and

processing a block of the target picture using the reference block.

2. The method according to clause 1, wherein the band-pass filter is a cosine windowed-sinc filter generated based on an ideal low-pass filter and a window function.

3. The method according to clause 2, wherein the cosine windowed-sinc filter is based on a kernel function $f(n) = \frac{fc}{r}\mathrm{Sinc}\left(\frac{fc}{r}n\right)\cos\left(\frac{\pi n}{L-1}\right)$, wherein $n = -\frac{L-1}{2}, \ldots, \frac{L-1}{2}$, wherein *fc* is a cutoff frequency of the cosine windowed-sinc filter, *L* is a kernel length, and *r* is a down-sampling ratio of the reference down-sampling.

4. The method according to clause 1, further comprising:

obtaining real filter coefficients of the band-pass filter;

determining a plurality of rounding directions for the real filter coefficients, respectively;

generating a plurality of combinations of rounded filter coefficients by rounding the real filter coefficients according to the plurality of rounding directions; and

selecting, among the plurality of combinations, a combination of rounded filter coefficients that causes a cost function to be minimum or maximum.

5. The method according to clause 4, wherein the cost function is associated with the rounded filter coefficients and a reference.

6. The method according to clause 5, wherein

the reference is the real filter coefficients of the band-pass filter before the rounding, a frequency response of an ideal filter, or a frequency response of a filter having a longer length than the band-pass filter.

7. The method according to clause 2, wherein the cosine windowed-sinc filter is an 8-tap filter, a 6-tap filter, or a 4-tap filter.

8. The method according to clause 1, wherein applying the band-pass filter to the reference picture comprises:

obtaining a luma sample or a chroma sample at a fractional sample position.

9. The method according to clause 1, further comprising:

determining the band-pass filter to be one of an 8-tap filter, a 6-tap filter, and a 4-tap filter, based on a ratio between the resolution of the reference picture and the resolution of the target picture.

10. The method according to clause 4, further comprising:

signaling the selected combination of rounded filter coefficients of the band-pass filter in at least one of Sequence Parameter Set (SPS), a Picture Parameter Set (PPS), an Adaptation Parameter Set (APS), and a slice header.

11. A system for performing motion compensation interpolation, comprising:

a memory storing a set of instructions; and

at least one processor configured to execute set of instructions to cause the system to perform:

in response to a target picture and a reference picture having different resolutions, applying a band-pass filter to the reference picture, to perform a motion compensated interpolation with reference down-sampling to generate a reference block; and

processing a block of the target picture using the reference block.

12. The system according to clause 11, wherein the band-pass filter is a cosine windowed-sinc filter generated based on an ideal low-pass filter and a window function.

13. The system according to clause 12, wherein the cosine windowed-sinc filter is based on a kernel function $f(n) = \frac{fc}{r}\mathrm{Sinc}\left(\frac{fc}{r}n\right)\cos\left(\frac{\pi n}{L-1}\right)$, wherein $n = -\frac{L-1}{2}, \ldots, \frac{L-1}{2}$, fc being a cutoff frequency of the cosine windowed-sinc filter, L being a kernel length, and r being a down-sampling ratio of the reference down-sampling.

14. The system according to clause 11, wherein at least one processor configured to execute set of instructions to cause the system to further perform:

obtaining real filter coefficients of the band-pass filter;

determining a plurality of rounding directions for the real filter coefficients, respectively;

generating a plurality of combinations of rounded filter coefficients by rounding the real filter coefficients according to the plurality of rounding directions; and

selecting, among the plurality of combinations, a combination of rounded filter coefficients that causes a cost function to be minimum or maximum.

15. The system according to clause 14, wherein the cost function is associated with the rounded filter coefficients and a reference.

16. The system according to clause 15, wherein

the reference is the real filter coefficients of the band-pass filter before the rounding, a frequency response of an ideal filter, or a frequency response of a filter having a longer length than the band-pass filter.

17. The system according to clause 12, wherein the cosine windowed-sinc filter is an 8-tap filter, a 6-tap filter, or a 4-tap filter.

18. The system according to clause 11, wherein applying the band-pass filter to the reference picture comprises:

obtaining a luma sample or a chroma sample at a fractional sample position.

19. The system according to clause 11, wherein at least one processor configured to execute set of instructions to cause the system to further perform:

determining the band-pass filter to be one of an 8-tap filter, a 6-tap filter, and a 4-tap filter, based on a ratio between the resolution of the reference picture and the resolution of the target picture.

20. A non-transitory computer readable medium that stores a set of instructions that is executable by at least one processor of a computer system to cause the computer system to perform a method for processing video content, the method comprising:

in response to a target picture and a reference picture having different resolutions, applying a band-pass filter to the reference picture, to perform a motion compensated interpolation with reference down-sampling to generate a reference block; and

processing a block of the target picture using the reference block.

[0137] It should be noted that, the relational terms herein such as “first” and “second” are used only to differentiate an entity or operation from another entity or operation, and do not require or imply any actual relationship or sequence between these entities or operations. Moreover, the words “comprising,” “having,” “containing,” and “including,” and other similar forms are intended to be equivalent in meaning and be open ended in that an item or items following any one of these words is not meant to be an exhaustive listing of such item or items, or meant to be limited to only the listed item or items.

[0138] As used herein, unless specifically stated otherwise, the term "or" encompasses all possible combinations, except where infeasible. For example, if it is stated that a database may include A or B, then, unless specifically stated otherwise or infeasible, the database may include A, or B, or A and B. As a second example, if it is stated that a database may include A, B, or C, then, unless specifically stated otherwise or infeasible, the database may include A, or B, or C, or A and B, or A and C, or B and C, or A and B and C.

[0139] It is appreciated that the above described embodiments can be implemented by hardware, or software (program codes), or a combination of hardware and software. If implemented by software, it may be stored in the above-described computer-readable media. The software, when executed by the processor can perform the disclosed methods. The computing units and other functional units described in this disclosure can be implemented by hardware, or software, or a combination of hardware and software. One of ordinary skill in the art will also understand that multiple ones of the above described modules/units may be combined as one module/unit, and each of the above described modules/units may be further divided into a plurality of sub-modules/sub-units.

[0140] In the foregoing specification, embodiments have been described with reference to numerous specific details that can vary from implementation to implementation. Certain adaptations and modifications of the described embodiments can be made. Other embodiments can be apparent to those skilled in the art from consideration of the specification and practice of the invention disclosed herein. It is intended that the specification and examples be considered as exemplary only, with a true scope and spirit of the invention being indicated by the following claims. It is also intended that the sequence of steps shown in figures are only for illustrative purposes and are not intended to be limited to any particular sequence of steps. As such, those skilled in the art can appreciate that these steps can be performed in a different order while implementing the same method.

[0141] In the drawings and specification, there have been disclosed exemplary embodiments. However, many variations and modifications can be made to these embodiments. Accordingly, although specific terms are employed, they are used in a generic and descriptive sense only and not for purposes of limitation.

WHAT IS CLAIMED IS:

1. A computer-implemented method for performing motion compensation interpolation, comprising:

in response to a target picture and a reference picture having different resolutions, applying a band-pass filter to the reference picture, to perform a motion compensated interpolation with reference down-sampling to generate a reference block; and

processing a block of the target picture using the reference block.

2. The method according to claim 1, wherein the band-pass filter is a cosine windowed-sinc filter generated based on an ideal low-pass filter and a window function.

3. The method according to claim 2, wherein the cosine windowed-sinc filter is based on a kernel function $f(n) = \frac{fc}{r}\text{Sinc}\left(\frac{fc}{r}n\right)\cos\left(\frac{\pi n}{L-1}\right)$, wherein $n = -\frac{L-1}{2}, \ldots, \frac{L-1}{2}$, wherein *fc* is a cutoff frequency of the cosine windowed-sinc filter, *L* is a kernel length, and *r* is a down-sampling ratio of the reference down-sampling.

4. The method according to claim 1, further comprising:

obtaining real filter coefficients of the band-pass filter;

determining a plurality of rounding directions for the real filter coefficients, respectively;

generating a plurality of combinations of rounded filter coefficients by rounding the real filter coefficients according to the plurality of rounding directions; and

selecting, among the plurality of combinations, a combination of rounded filter coefficients that causes a cost function to be minimum or maximum.

5. The method according to claim 4, wherein the cost function is associated with the rounded filter coefficients and a reference.

6. The method according to claim 5, wherein

the reference is the real filter coefficients of the band-pass filter before the rounding, a frequency response of an ideal filter, or a frequency response of a filter having a longer length than the band-pass filter.

7. The method according to claim 2, wherein the cosine windowed-sinc filter is an 8-tap filter, a 6-tap filter, or a 4-tap filter.

8. The method according to claim 1, wherein applying the band-pass filter to the reference picture comprises:

obtaining a luma sample or a chroma sample at a fractional sample position.

9. The method according to claim 1, further comprising:

determining the band-pass filter to be one of an 8-tap filter, a 6-tap filter, and a 4-tap filter, based on a ratio between the resolution of the reference picture and the resolution of the target picture.

10. The method according to claim 4, further comprising:

signaling the selected combination of rounded filter coefficients of the band-pass filter in at least one of Sequence Parameter Set (SPS), a Picture Parameter Set (PPS), an Adaptation Parameter Set (APS), and a slice header.

11. A system for performing motion compensation interpolation, comprising:

a memory storing a set of instructions; and

at least one processor configured to execute set of instructions to cause the system to perform:

in response to a target picture and a reference picture having different resolutions, applying a band-pass filter to the reference picture, to perform a motion compensated interpolation with reference down-sampling to generate a reference block; and

processing a block of the target picture using the reference block.

12. The system according to claim 11, wherein the band-pass filter is a cosine windowed-sinc filter generated based on an ideal low-pass filter and a window function.

13. The system according to claim 12, wherein the cosine windowed-sinc filter is based on a kernel function $f(n) = \frac{fc}{r}\mathrm{Sinc}\left(\frac{fc}{r}n\right)\cos\left(\frac{\pi n}{L-1}\right)$, wherein $n = -\frac{L-1}{2}, \ldots, \frac{L-1}{2}$, fc being a cutoff frequency of the cosine windowed-sinc filter, L being a kernel length, and r being a down-sampling ratio of the reference down-sampling.

14. The system according to claim 11, wherein at least one processor configured to execute set of instructions to cause the system to further perform:

obtaining real filter coefficients of the band-pass filter;

determining a plurality of rounding directions for the real filter coefficients, respectively;

generating a plurality of combinations of rounded filter coefficients by rounding the real filter coefficients according to the plurality of rounding directions; and

selecting, among the plurality of combinations, a combination of rounded filter coefficients that causes a cost function to be minimum or maximum.

15. The system according to claim 14, wherein the cost function is associated with the rounded filter coefficients and a reference.

16. The system according to claim 15, wherein

the reference is the real filter coefficients of the band-pass filter before the rounding, a frequency response of an ideal filter, or a frequency response of a filter having a longer length than the band-pass filter.

17. The system according to claim 12, wherein the cosine windowed-sinc filter is an 8-tap filter, a 6-tap filter, or a 4-tap filter.

18. The system according to claim 11, wherein applying the band-pass filter to the reference picture comprises:

obtaining a luma sample or a chroma sample at a fractional sample position.

19. The system according to claim 11, wherein at least one processor configured to execute set of instructions to cause the system to further perform:

determining the band-pass filter to be one of an 8-tap filter, a 6-tap filter, and a 4-tap filter, based on a ratio between the resolution of the reference picture and the resolution of the target picture.

20. A non-transitory computer readable medium that stores a set of instructions that is executable by at least one processor of a computer system to cause the computer system to perform a method for processing video content, the method comprising:

in response to a target picture and a reference picture having different resolutions, applying a band-pass filter to the reference picture, to perform a motion compensated interpolation with reference down-sampling to generate a reference block; and

processing a block of the target picture using the reference block.

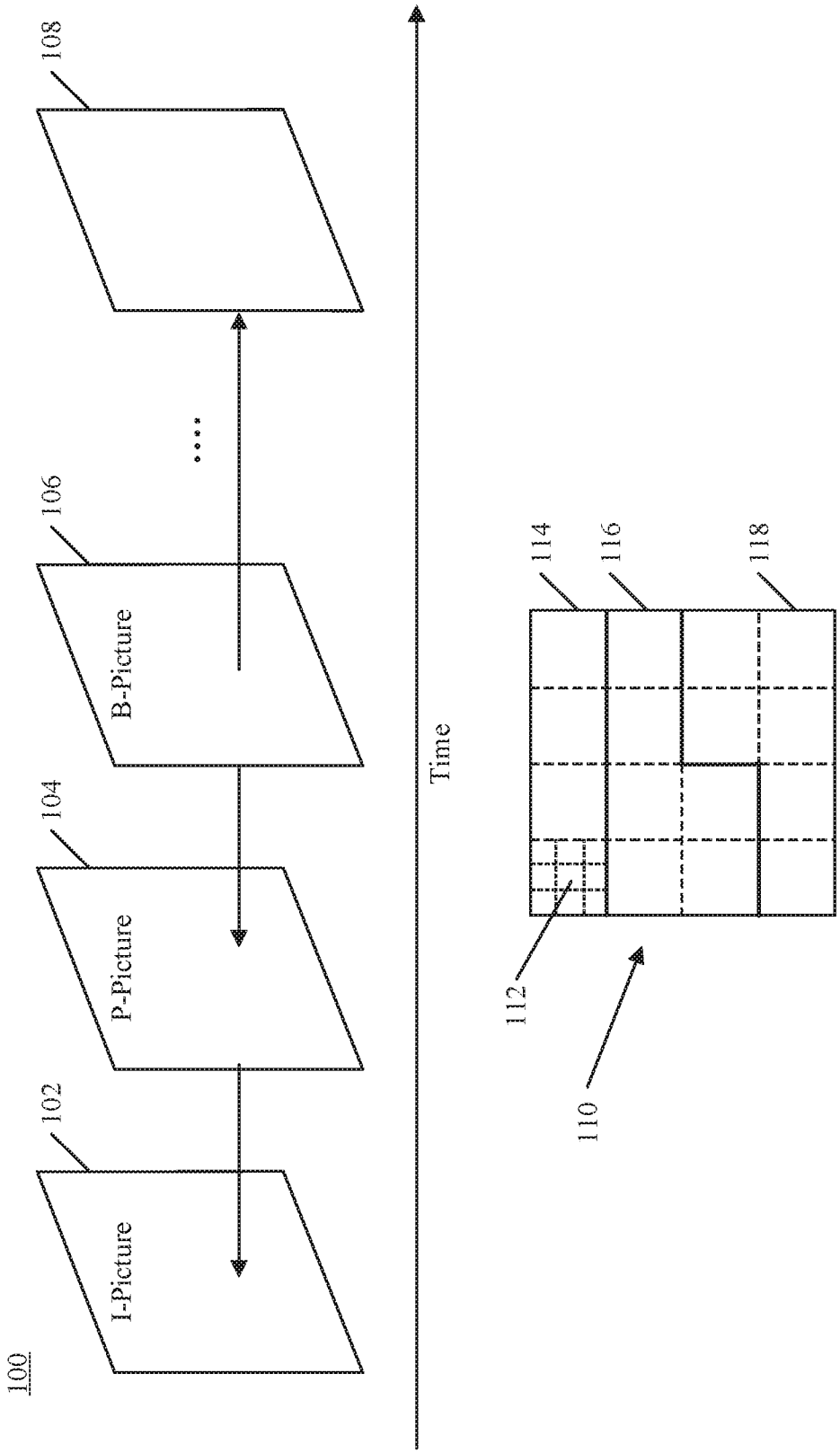

100
I-Picture
102
P-Picture
104
B-Picture
106
108
Time
110
112
114
116
118

FIG. 1

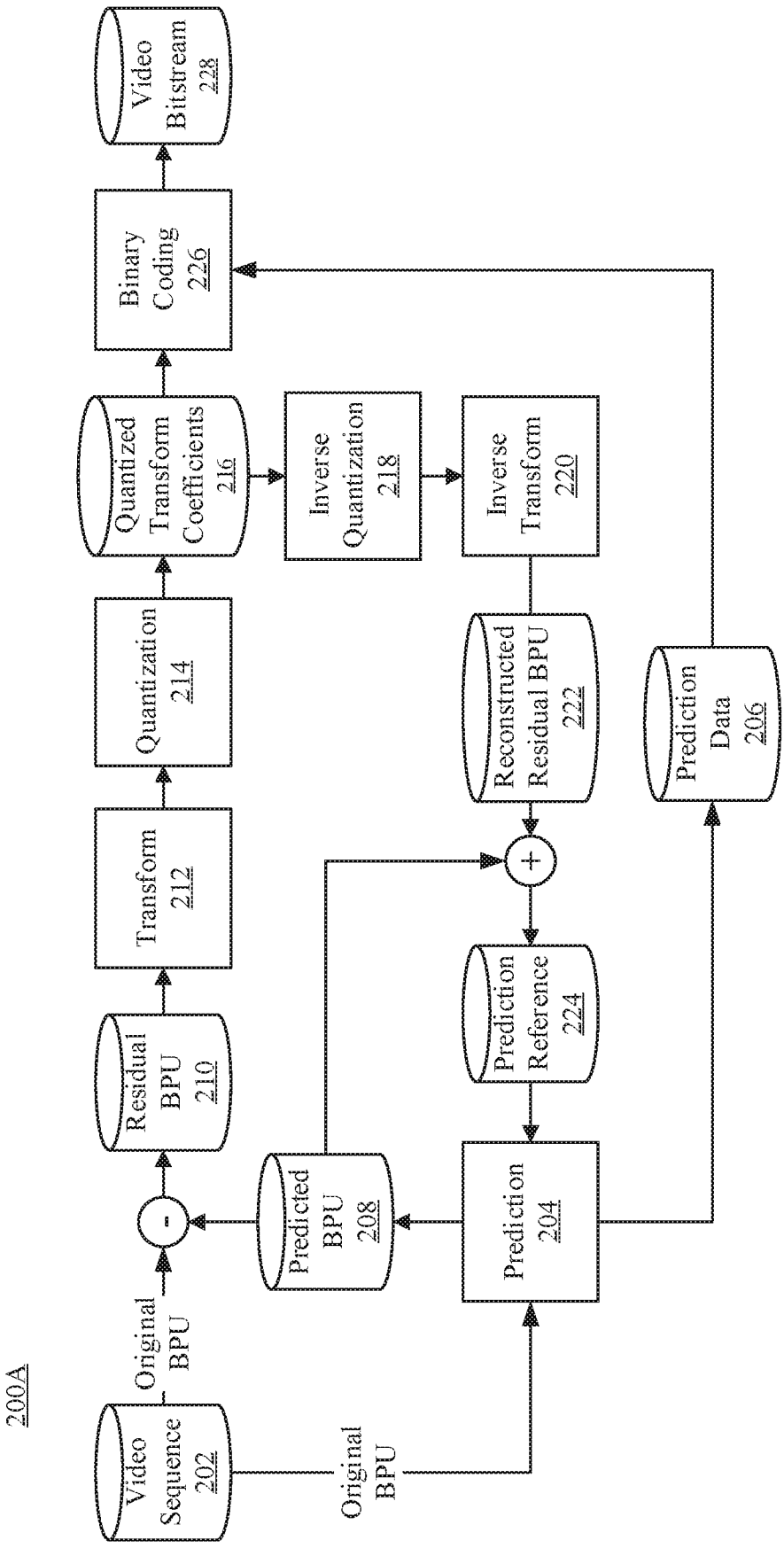

200A
Video Sequence 202
Original BPU
Original BPU
-
Residual BPU 210
Transform 212
Quantization 214
Quantized Transform Coefficients 216
Inverse Quantization 218
Inverse Transform 220
Reconstructed Residual BPU 222
+
Predicted BPU 208
Prediction 204
Prediction Reference 224
Prediction Data 206
Binary Coding 226
Video Bitstream 228

FIG. 2A

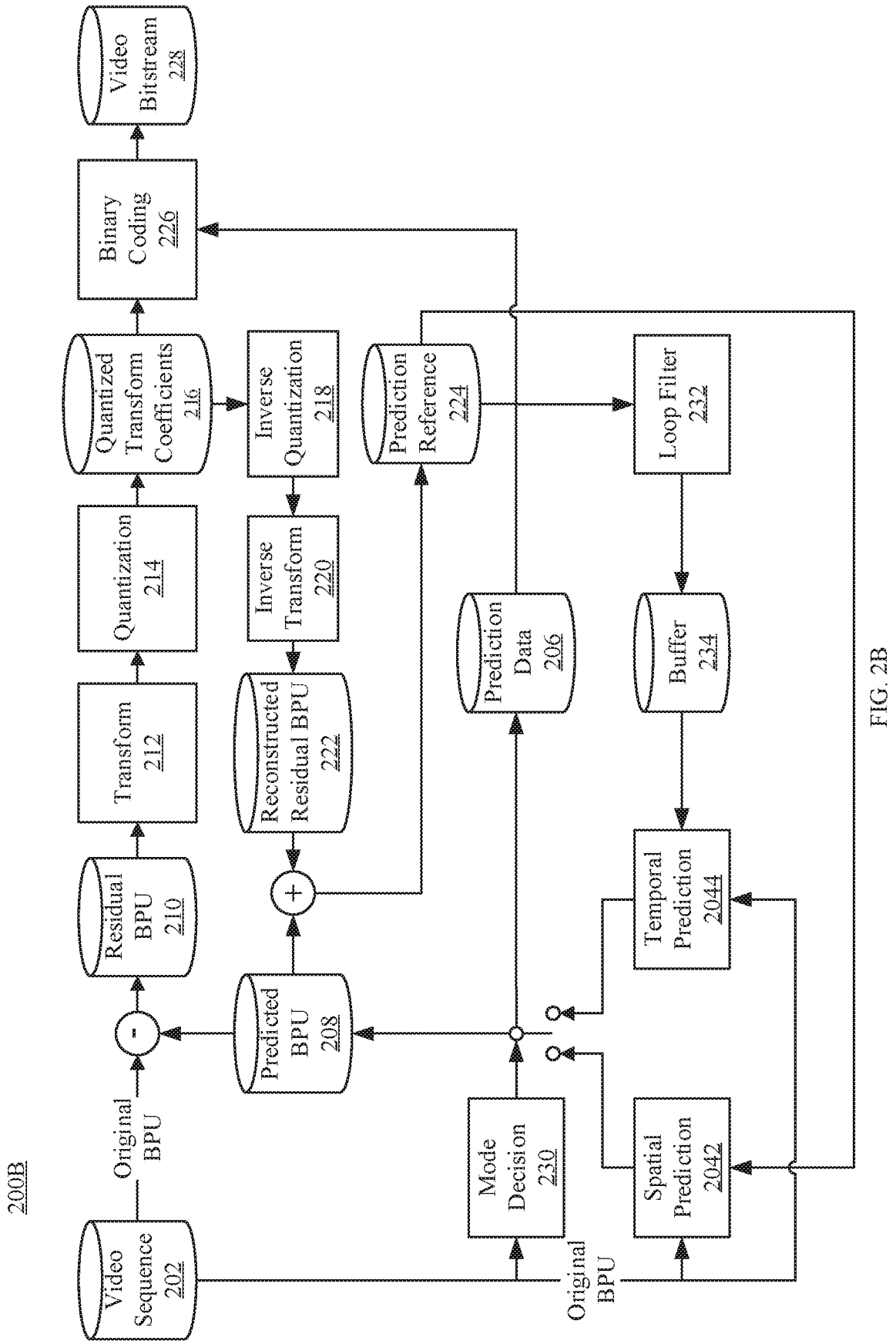

200B
Video Sequence 202
Original BPU
Mode Decision 230
Spatial Prediction 2042
Temporal Prediction 2044
Predicted BPU 208
Residual BPU 210
Transform 212
Quantization 214
Quantized Transform Coefficients 216
Binary Coding 226
Video Bitstream 228
Inverse Quantization 218
Inverse Transform 220
Reconstructed Residual BPU 222
Prediction Data 206
Prediction Reference 224
Loop Filter 232
Buffer 234
Original BPU

FIG. 2B

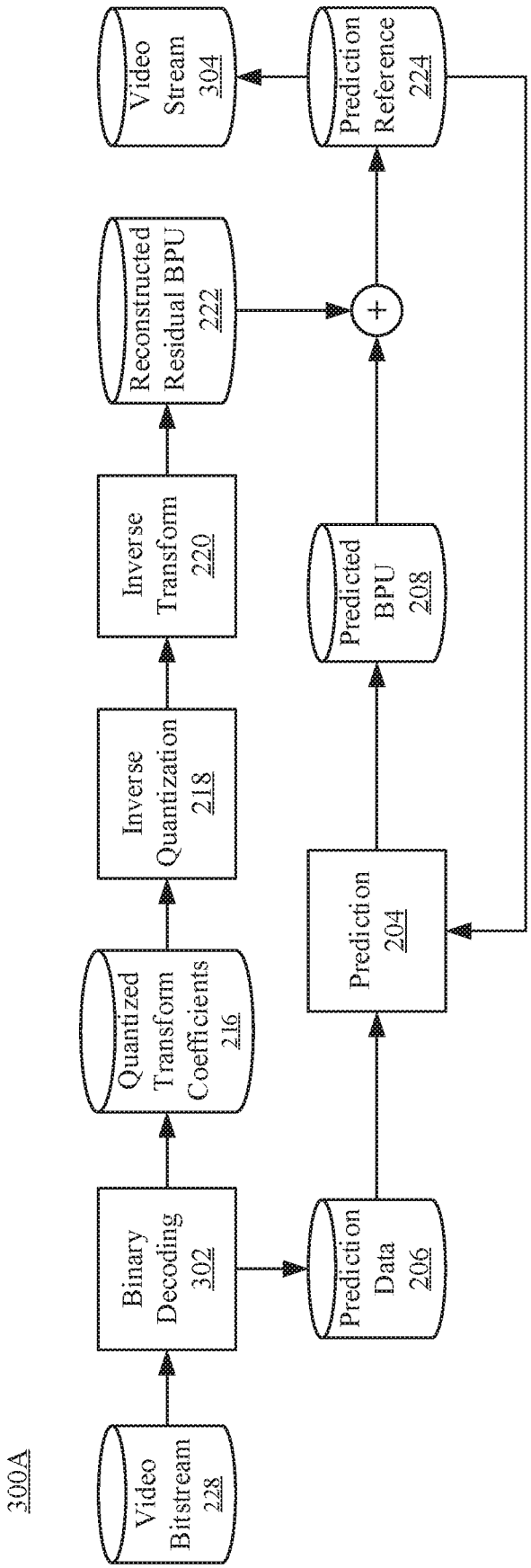

300A
Video Bitstream 228
Binary Decoding 302
Quantized Transform Coefficients 216
Inverse Quantization 218
Inverse Transform 220
Reconstructed Residual BPU 222
Video Stream 304
Prediction Data 206
Prediction 204
Predicted BPU 208
Prediction Reference 224
+

FIG. 3A

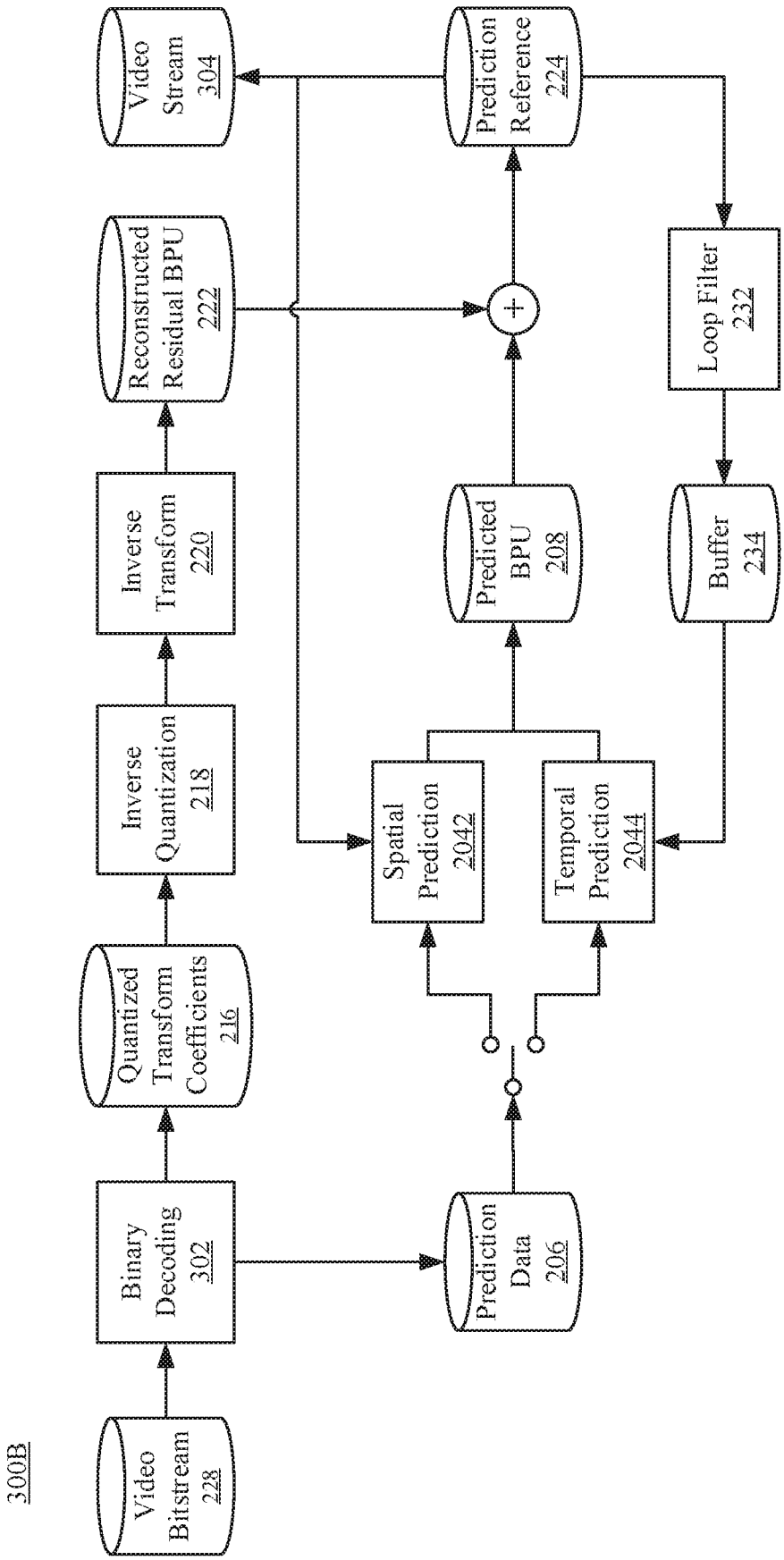

300B
Video Bitstream 228
Binary Decoding 302
Quantized Transform Coefficients 216
Inverse Quantization 218
Inverse Transform 220
Reconstructed Residual BPU 222
Video Stream 304
Prediction Data 206
Predicted BPU 208
Spatial Prediction 2042
Temporal Prediction 2044
Prediction Reference 224
Loop Filter 232
Buffer 234

FIG. 3B

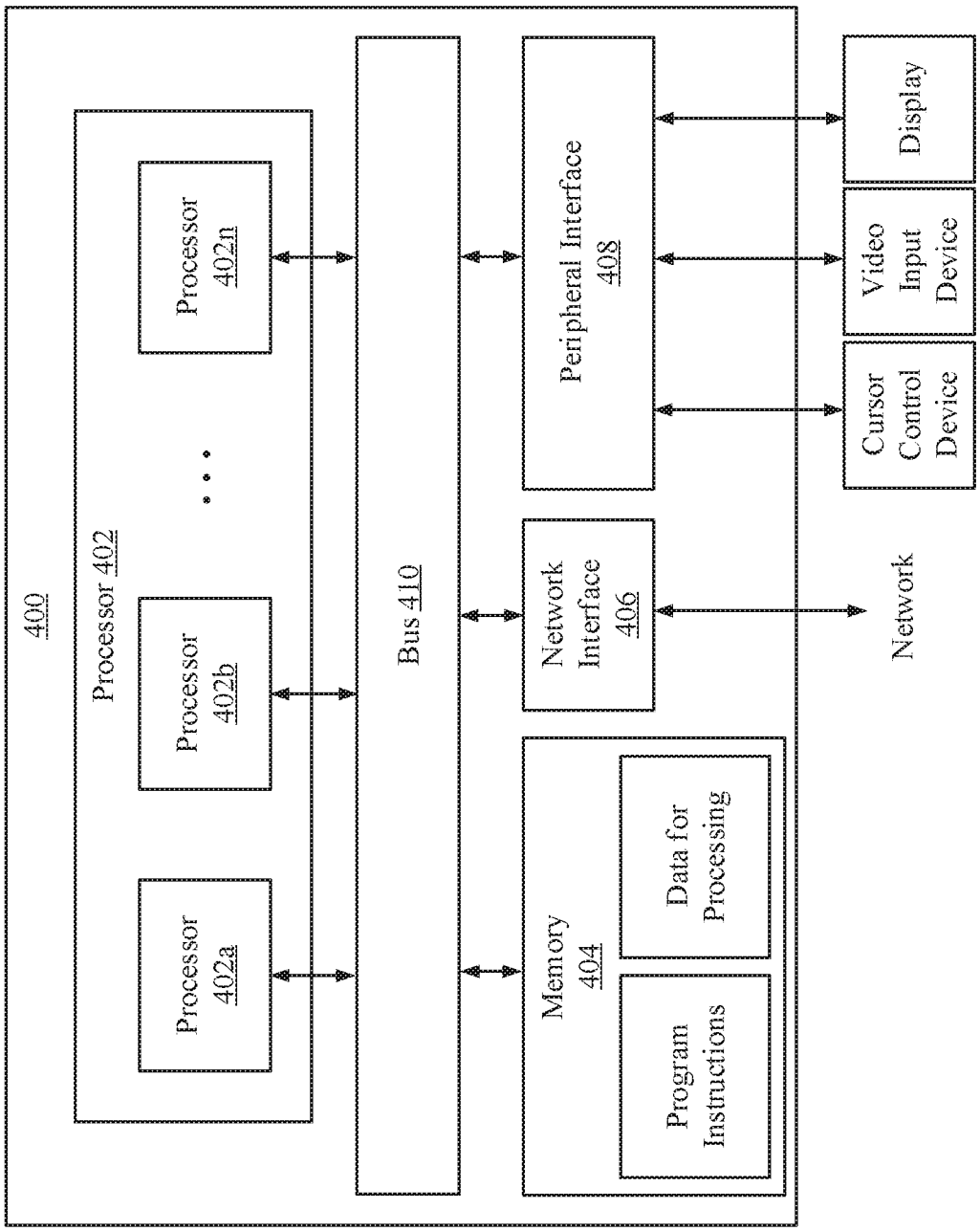
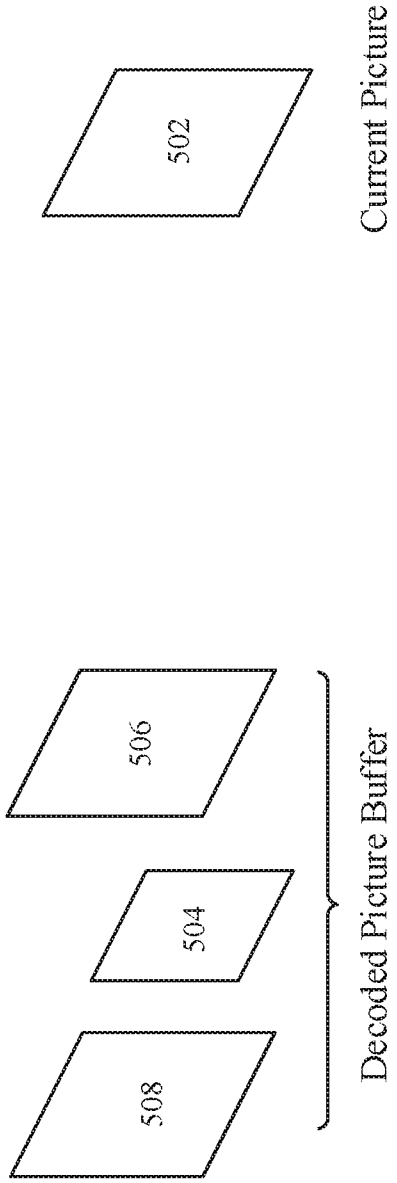

400
Processor 402
Processor 402a
Processor 402b
Processor 402n
Bus 410
Memory 404
Program Instructions
Data for Processing
Network Interface 406
Peripheral Interface 408
Network
Display
Video Input Device
Cursor Control Device

FIG. 4



500

FIG. 5

Table 6: 6-tap DCT-based interpolation filter coefficients for 4×4 luma block in VVC

Fractional sample position p	6-tap interpolation filter coefficients for 4x4 luma block in VVC					
	p0	p1	p2	p3	p4	p5
0	0	0	64	0	0	0
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

FIG. 6

Table 7: 8-tap interpolation filter coefficients for luma in VVC

Fractional sample position p	8-tap interpolation filter coefficients for luma in VVC							
	p0	p1	p2	p3	p4	p5	p6	p7
0	0	0	0	64	0	0	0	0
1	0	1	-3	63	4	-2	0	1
2	-1	2	-5	62	8	-3	-1	2
3	-1	3	-8	60	13	-4	-1	3
4	-1	4	-10	58	17	-5	-1	4
5	-1	4	-11	52	26	-8	-1	4
6	-1	3	-9	47	31	-10	-1	3
7	-1	4	-11	45	34	-10	-1	4
8	-1	4	-11	40	40	-11	-1	4
9	-1	4	-10	34	45	-11	-1	4
10	-1	4	-10	31	47	-9	-1	4
11	-1	3	-8	26	52	-11	-1	3
12	0	1	-5	17	58	-10	0	1
13	0	1	-4	13	60	-8	0	1
14	0	1	-3	8	62	-5	0	1
15	0	1	-2	4	63	-3	0	1

FIG. 7

Table 8: 4-tap 32-phase interpolation filter coefficients for chroma in VVC

Fractional sample position p	4-tap interpolation filter coefficients for chroma in VVC			
	p0	p1	p2	p3
0	0	64	0	0
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

FIG. 8

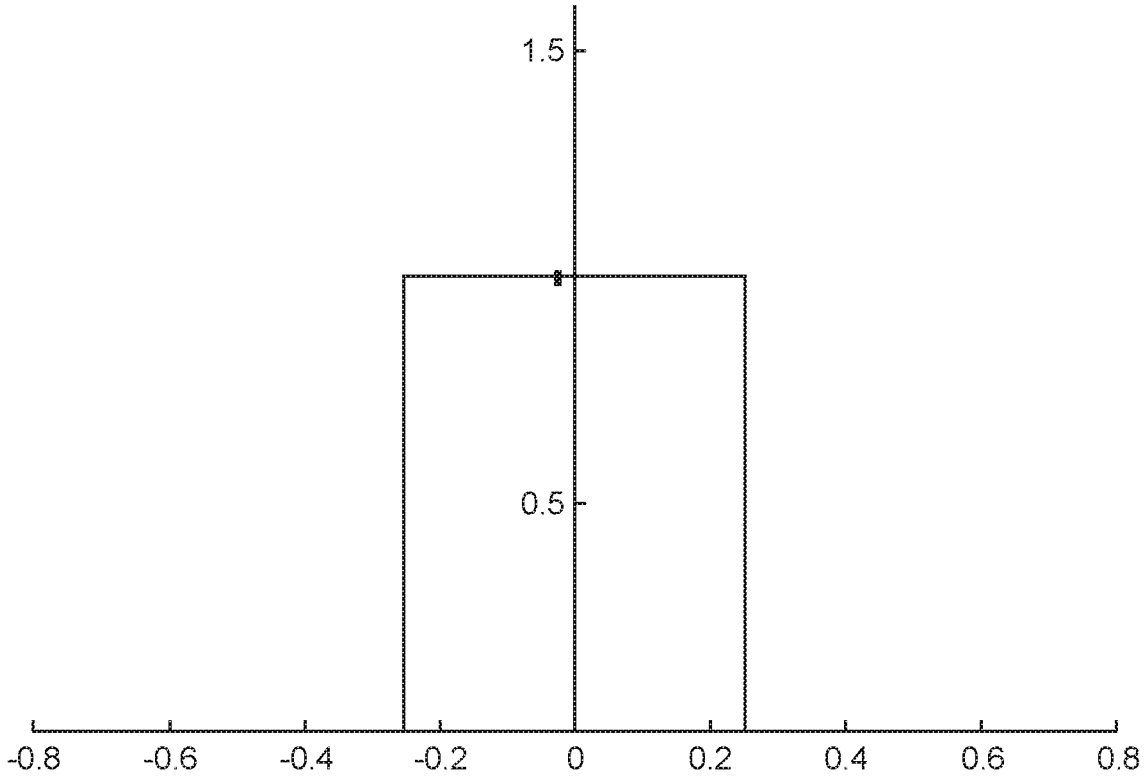


FIG. 9

Table 10: 12-tap cosine-windowed-sinc interpolation filter coefficients for 2:1 down-sampling

Fractional sample position p	Interpolation filter coefficients for 2:1 down-sampling											
	p0	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11
0	2	-3	-9	6	39	58	39	6	-9	-3	2	0
1	2	-3	-9	4	38	58	43	7	-9	-4	1	0
2	2	-2	-9	2	35	58	44	9	-8	-4	1	0
3	1	-2	-9	1	34	58	46	11	-8	-5	1	0
4	1	-1	-8	-1	31	57	47	13	-7	-5	1	0
5	1	-1	-8	-2	29	56	49	15	-7	-6	1	1
6	1	0	-8	-3	26	55	51	17	-7	-6	1	1
7	1	0	-7	-4	24	54	52	19	-6	-7	1	1
8	1	0	-7	-5	22	53	53	22	-5	-7	0	1
9	1	1	-7	-6	19	52	54	24	-4	-7	0	1
10	1	1	-6	-7	17	51	55	26	-3	-8	0	1
11	1	1	-6	-7	15	49	56	29	-2	-8	-1	1
12	0	1	-5	-7	13	47	57	31	-1	-8	-1	1
13	0	1	-5	-8	11	46	58	34	1	-9	-2	1
14	0	1	-4	-8	9	44	58	35	2	-9	-2	2
15	0	1	-4	-9	7	43	58	38	4	-9	-3	2

FIG. 10

Table 11: 12-tap cosine-windowed-sinc interpolation filter coefficients for 1.5:1 down-sampling

Fractional sample position p	**Interpolation filter coefficients for 1.5 : 1 down-sampling**											
	p0	**p1**	**p2**	**p3**	**p4**	**p5**	**p6**	**p7**	**p8**	**p9**	**p10**	**p11**
0	0	5	-6	-10	37	76	37	-10	-6	5	0	0
1	0	5	-4	-11	33	76	40	-9	-7	5	0	0
2	-1	5	-3	-12	29	75	45	-7	-8	5	0	0
3	-1	4	-2	-13	25	75	48	-5	-9	5	1	0
4	-1	4	-1	-13	22	73	52	-3	-10	4	1	0
5	-1	4	0	-13	18	72	55	-1	-11	4	2	-1
6	-1	4	1	-13	14	70	59	2	-12	3	2	-1
7	-1	3	1	-13	11	68	62	5	-12	3	2	-1
8	-1	3	2	-13	8	65	65	8	-13	2	3	-1
9	-1	2	3	-12	5	62	68	11	-13	1	3	-1
10	-1	2	3	-12	2	59	70	14	-13	1	4	-1
11	-1	2	4	-11	-1	55	72	18	-13	0	4	-1
12	0	1	4	-10	-3	52	73	22	-13	-1	4	-1
13	0	1	5	-9	-5	48	75	25	-13	-2	4	-1
14	0	0	5	-8	-7	45	75	29	-12	-3	5	-1
15	0	0	5	-7	-9	40	76	33	-11	-4	5	0

FIG. 11

Table 12: Exemplary luma sample interpolation filtering process for reference down-sampling

The predicted luma sample value $predSampleLX_L$ is derived as follows:

– The sample array temp[n] with n = 0……11, is derived as follows:

$$temp[\,n\,] = \sum_{i=0}^{7} f_L[\,xFrac_L\,][\,i\,] * refPicLX_L[\,xInt_i\,][\,yInt_n\,] \gg shift1$$

– The predicted luma sample value $predSampleLX_L$ is derived as follows:

$$predSampleLX_L = \sum_{i=0}^{7} f_L[\,yFrac_L\,][\,i\,] * temp[\,i\,] \gg shift2$$

FIG. 12

Table 13: Exemplary chroma sample interpolation filtering process for reference down-sampling

The predicted chroma sample value $predSampleLX_C$ is derived as follows:

– The sample array temp[n] with n = 0.....11, is derived as follows:

$$temp[n] = \sum_{i=0}^{3} f_C[xFrac_C][i] * refPicLX_C[xInt_i][yInt_n] >> shift1$$

– The predicted chroma sample value $predSampleLX_C$ is derived as follows:

$$predSampleLX_C = \sum_{i=0}^{7} f_C[yFrac_C][i] * temp[i] >> shift2$$

FIG. 13

Table 14: Exemplary luma sample interpolation filtering process for reference down-sampling

– The sample array temp[n] with n = 0……11, is derived as follows:

$$temp[n] = \sum_{i=0}^{7} f_L[xFrac_L][i] * refPicLX_L[xInt_i][yInt_n] >> (shift1 + 1)$$

– The predicted luma sample value $predSampleLX_L$ is derived as follows:

$$predSampleLX_L = \sum_{i=0}^{7} f_L[yFrac_L][i] * temp[i] >> (shift2 + 1)$$

FIG. 14

Table 15: Exemplary chroma sample interpolation filtering process for reference down-sampling

– The sample array temp[n] with n = 0.....11, is derived as follows:

$$temp[n] = \sum_{i=0}^{3} f_C[xFrac_C][i] * refPicLX_C[xInt_i][yInt_n] >> (shift1 + 1)$$

– The predicted chroma sample value predSampleLX$_C$ is derived as follows:

$$predSampleLX_C = \sum_{i=0}^{7} f_C[yFrac_C][i] * temp[i] >> (shift2 + 1)$$

FIG. 15

Table 16: Exemplary luma sample interpolation filtering process for reference down-sampling

– The sample array temp[n] with n = 0……11, is derived as follows:

$$temp[n] = \sum_{i=0}^{7} f_L[xFrac_L][i] * refPicLX_L[xInt_i][yInt_n] >> shift1$$

– The predicted luma sample value $predSampleLX_L$ is derived as follows:

$$predSampleLX_L = \sum_{i=0}^{7} f_L[yFrac_L][i] * temp[i] >> (shift2 + 2)$$

FIG. 16

Table 17: Exemplary chroma sample interpolation filtering process for reference down-sampling

– The sample array temp[n] with n = 0……11, is derived as follows:

$$temp[n] = \sum_{i=0}^{3} f_C[xFrac_C][i] * refPicLX_C[xInt_i][yInt_n] >> shift1$$

– The predicted chroma sample value predSampleLX$_C$ is derived as follows:

$$predSampleLX_C = \sum_{i=0}^{7} f_C[yFrac_C][i] * temp[i] >> (shift2 + 2)$$

FIG. 17

Table 18: Exemplary chroma sample interpolation filtering process for reference down-sampling

– The variables $xInt_C$, $yInt_C$, $xFrac_C$ and $yFrac_C$ are derived as follows:

$xInt_C = refx_C >> 5$

$yInt_C = refy_C >> 5$

$xFrac_C = (refy_C \& 31) >> 1$

$yFrac_C = (refy_C \& 31) >> 1$

FIG. 18

Table 19: 8-tap filter for MC interpolation with reference down-sampling at ratio 2:1

Fractional sample position p	8-tap filter for MC interpolation with reference down-sampling at ratio 2:1							
	p0	p1	p2	p3	p4	p5	p6	p7
0	-2	2	18	28	18	2	-2	0
1	-2	1	17	28	19	3	-2	0
2	-2	1	16	28	20	4	-3	0
3	-2	0	15	28	21	4	-2	0
4	-2	0	14	27	22	5	-2	0
5	-2	-1	13	27	23	6	-2	0
6	-1	-1	12	26	24	7	-2	-1
7	-1	-1	10	26	25	8	-2	-1
8	-1	-2	10	25	25	10	-2	-1
9	-1	-2	8	25	26	10	-1	-1
10	-1	-2	7	24	27	12	-1	-2
11	0	-2	6	23	27	13	-1	-2
12	-1	-2	5	22	28	14	0	-2
13	0	-2	4	21	28	15	0	-2
14	0	-3	4	20	28	16	1	-2
15	0	-2	3	19	28	17	1	-2

FIG. 19

Table 20: 8-tap filter for MC interpolation with reference down-sampling at ratio 1.5:1

Fractional sample position p	8-tap filter for MC interpolation with reference down-sampling at ratio 1.5:1							
	p0	p1	p2	p3	p4	p5	p6	p7
0	-2	-4	18	40	18	-4	-2	0
1	-1	-5	16	39	21	-4	-2	0
2	-1	-5	14	39	22	-3	-2	0
3	0	-5	12	38	24	-2	-3	0
4	0	-5	10	38	26	-2	-3	0
5	0	-5	8	37	28	0	-4	0
6	0	-5	7	35	30	1	-4	0
7	0	-5	5	34	32	2	-4	0
8	0	-5	4	33	33	4	-5	0
9	0	-4	2	32	34	5	-5	0
10	0	-4	1	30	35	7	-5	0
11	0	-4	0	28	37	8	-5	0
12	0	-3	-2	26	38	10	-5	0
13	0	-3	-2	24	38	12	-5	0
14	0	-2	-3	22	39	14	-5	-1
15	0	-2	-4	21	39	16	-5	-1

FIG. 20

Table 21: 8-tap filter for MC interpolation with reference down-sampling at ratio 2:1 with 32 phases

Fractional sample position p	8-tap filter for MC interpolation with reference down-sampling at ratio 2:1							
	p0	p1	p2	p3	p4	p5	p6	p7
0	-2	2	18	28	18	2	-2	0
1	-2	2	17	28	19	2	-2	0
2	-2	1	17	28	19	3	-2	0
3	-2	1	16	28	20	3	-2	0
4	-2	1	16	28	20	4	-3	0
5	-2	0	15	28	21	4	-2	0
6	-2	0	15	28	21	4	-2	0
7	-2	0	14	27	22	5	-2	0
8	-2	0	14	27	22	5	-2	0
9	-2	0	13	27	23	6	-2	-1
10	-2	-1	13	27	23	6	-2	0
11	-1	-1	12	27	23	7	-2	-1
12	-1	-1	12	26	24	7	-2	-1
13	-1	-1	11	26	24	8	-2	-1
14	-1	-1	10	26	25	8	-2	-1
15	-1	-2	10	26	25	9	-2	-1
16	-1	-2	10	25	25	10	-2	-1
17	-1	-2	9	25	26	10	-2	-1
18	-1	-2	8	25	26	10	-1	-1
19	-1	-2	8	24	26	11	-1	-1
20	-1	-2	7	24	27	12	-1	-2
21	-1	-2	7	24	27	12	-1	-2
22	0	-2	6	23	27	13	-1	-2
23	-1	-2	6	23	27	13	0	-2
24	-1	-2	5	22	28	14	0	-2
25	-1	-2	5	22	28	14	0	-2
26	0	-2	4	21	28	15	0	-2
27	0	-2	4	21	28	15	0	-2
28	0	-3	4	20	28	16	1	-2
29	0	-3	3	20	28	17	1	-2
30	0	-2	3	19	28	17	1	-2
31	0	-3	2	19	28	18	2	-2

FIG. 21

Table 22: 8-tap filter for MC interpolation with reference down-sampling at ratio 1.5:1 with 32 phases

Fractional sample position p	8-tap filter for MC interpolation with reference down-sampling at ratio 1.5:1							
	p0	p1	p2	p3	p4	p5	p6	p7
0	-2	-4	18	40	18	-4	-2	0
1	-1	-4	17	39	19	-4	-2	0
2	-1	-5	16	39	21	-4	-2	0
3	-1	-5	15	39	21	-3	-2	0
4	-1	-5	14	39	22	-3	-2	0
5	-1	-5	13	39	23	-3	-2	0
6	0	-5	12	38	24	-2	-3	0
7	0	-5	11	38	25	-2	-3	0
8	0	-5	10	38	26	-2	-3	0
9	0	-5	9	37	27	-1	-3	0
10	0	-5	8	37	28	0	-4	0
11	0	-5	8	36	29	0	-4	0
12	0	-5	7	35	30	1	-4	0
13	0	-5	6	35	31	1	-4	0
14	0	-5	5	34	32	2	-4	0
15	0	-5	4	34	32	3	-4	0
16	0	-5	4	33	33	4	-5	0
17	0	-4	3	32	34	4	-5	0
18	0	-4	2	32	34	5	-5	0
19	0	-4	1	31	35	6	-5	0
20	0	-4	1	30	35	7	-5	0
21	0	-4	0	29	36	8	-5	0
22	0	-4	0	28	37	8	-5	0
23	0	-3	-1	27	37	9	-5	0
24	0	-3	-2	26	38	10	-5	0
25	0	-3	-2	25	38	11	-5	0
26	0	-3	-2	24	38	12	-5	0
27	0	-2	-3	23	39	13	-5	-1
28	0	-2	-3	22	39	14	-5	-1
29	0	-2	-3	21	39	15	-5	-1
30	0	-2	-4	21	39	16	-5	-1
31	0	-2	-4	19	39	17	-4	-1

FIG. 22

Table 23: 6-tap filter coefficients for luma 4×4 block MC interpolation with reference down-sampling ratio 2:1 with 16 phases

Fractional sample position p	6-tap filter coefficients for luma 4×4 block MC interpolation with reference down-sampling with ratio 2:1					
	p0	p1	p2	p3	p4	p5
0	1	17	28	17	1	0
1	1	16	27	18	2	0
2	1	14	27	19	3	0
3	0	13	27	20	4	0
4	0	12	27	21	4	0
5	0	11	26	22	5	0
6	0	10	26	23	6	-1
7	0	9	25	24	7	-1
8	-1	8	25	25	8	-1
9	-1	7	24	25	9	0
10	-1	6	23	26	10	0
11	0	5	22	26	11	0
12	0	4	21	27	12	0
13	0	4	20	27	13	0
14	0	3	19	27	14	1
15	0	2	18	27	16	1

FIG. 23

Table 24: 6-tap filter coefficients for luma 4×4 block MC interpolation with reference down-sampling ratio 1.5:1 with 16 phases

Fractional sample position p	**6-tap filter coefficients for luma 4×4 block MC interpolation with reference down-sampling with ratio 1.5:1**					
	p0	**p1**	**p2**	**p3**	**p4**	**p5**
0	-3	16	38	16	-3	0
1	-3	14	37	18	-2	0
2	-3	12	37	20	-2	0
3	-3	11	36	22	-2	0
4	-3	9	36	24	-1	-1
5	-3	7	35	26	0	-1
6	-3	6	34	28	1	-2
7	-3	4	33	30	2	-2
8	-2	3	31	31	3	-2
9	-2	2	30	33	4	-3
10	-2	1	28	34	6	-3
11	-1	0	26	35	7	-3
12	-1	-1	24	36	9	-3
13	-1	-2	23	37	11	-4
14	-1	-3	21	38	13	-4
15	-1	-3	19	38	15	-4

FIG. 24

Table 25: 8-tap filter for MC interpolation with reference down-sampling at ratio 2:1

Fractional sample position p	8-tap filter for MC interpolation with reference down-sampling at ratio 2:1							
	p0	p1	p2	p3	p4	p5	p6	p7
0	-2	2	18	28	18	2	-2	0
1	-2	1	17	28	19	3	-2	0
2	-2	1	16	28	20	4	-3	0
3	-2	0	15	28	21	4	-2	0
4	-2	0	14	27	22	5	-2	0
5	-2	-1	13	27	23	6	-2	0
6	-2	-1	12	27	24	7	-2	-1
7	-1	-1	10	26	25	8	-2	-1
8	-1	-2	10	25	25	10	-2	-1
9	-1	-2	8	25	26	10	-1	-1
10	-1	-2	7	24	27	12	-1	-2
11	0	-2	6	23	27	13	-1	-2
12	0	-2	5	22	27	14	0	-2
13	0	-2	4	21	28	15	0	-2
14	0	-3	4	20	28	16	1	-2
15	0	-2	3	19	28	17	1	-2

FIG. 25

Table 26: 8-tap filter for MC interpolation with reference down-sampling at ratio 1.5:1

Fractional sample position p	8-tap filter for MC interpolation with reference down-sampling at ratio 1.5:1							
	p0	p1	p2	p3	p4	p5	p6	p7
0	-2	-4	18	40	18	-4	-2	0
1	-1	-5	16	39	21	-4	-2	0
2	-1	-5	14	39	22	-3	-2	0
3	0	-5	12	38	24	-2	-3	0
4	0	-5	10	38	26	-2	-3	0
5	0	-5	8	37	28	0	-4	0
6	0	-5	7	35	30	1	-4	0
7	0	-5	5	34	32	2	-4	0
8	0	-5	4	33	33	4	-5	0
9	0	-4	2	32	34	5	-5	0
10	0	-4	1	30	35	7	-5	0
11	0	-4	0	28	37	8	-5	0
12	0	-3	-2	26	38	10	-5	0
13	0	-3	-2	24	38	12	-5	0
14	0	-2	-3	22	39	14	-5	-1
15	0	-2	-4	21	39	16	-5	-1

FIG. 26

Table 27: 6-tap filter for MC interpolation with reference down-sampling at ratio 2:1 with 16 phases

Fractional sample position p	6-tap filter coefficients for luma 4x4 block MC interpolation with reference down-sampling with ratio 2:1					
	p0	p1	p2	p3	p4	p5
0	1	17	28	17	1	0
1	1	16	27	18	2	0
2	1	14	27	19	3	0
3	0	13	27	20	4	0
4	0	12	27	21	4	0
5	0	11	26	22	5	0
6	0	10	26	23	6	-1
7	0	9	25	24	7	-1
8	-1	8	25	25	8	-1
9	-1	7	24	25	9	0
10	-1	6	23	26	10	0
11	0	5	22	26	11	0
12	0	4	21	27	12	0
13	0	4	20	27	13	0
14	0	3	19	27	14	1
15	0	2	18	27	16	1

FIG. 27

Table 28: 6-tap filter for MC interpolation with reference down-sampling at ratio 1.5:1 with 16 phases

Fractional sample position p	6-tap filter coefficients for luma 4x4 block MC interpolation with reference down-sampling with ratio 1.5:1					
	p0	p1	p2	p3	p4	p5
0	-3	16	38	16	-3	0
1	-3	15	37	18	-3	0
2	-3	13	37	20	-2	-1
3	-3	11	37	22	-2	-1
4	-3	9	36	24	-1	-1
5	-3	7	35	26	0	-1
6	-3	6	34	28	1	-2
7	-3	4	33	30	2	-2
8	-2	3	31	31	3	-2
9	-2	2	30	33	4	-3
10	-2	1	28	34	6	-3
11	-1	0	26	35	7	-3
12	-1	-1	24	36	9	-3
13	-1	-2	22	37	11	-3
14	-1	-2	20	37	13	-3
15	0	-3	18	37	15	-3

FIG. 28

Table 29: 4-tap filter for MC interpolation with reference down-sampling at ratio 2:1 with 32-phases

Fractional sample position p	4-tap filter for MC interpolation with reference down-sampling at ratio 2:1			
	p0	p1	p2	p3
0	16	32	16	0
1	15	32	17	0
2	14	32	18	0
3	13	32	18	1
4	13	31	19	1
5	12	31	20	1
6	11	31	21	1
7	11	31	21	1
8	10	30	22	2
9	9	30	23	2
10	9	30	23	2
11	8	29	24	3
12	7	29	25	3
13	7	28	25	4
14	6	28	26	4
15	6	27	26	5
16	5	27	27	5
17	5	26	27	6
18	4	26	28	6
19	4	25	28	7
20	3	25	29	7
21	3	24	29	8
22	2	23	30	9
23	2	23	30	9
24	2	22	30	10
25	1	21	31	11
26	1	21	31	11
27	1	20	31	12
28	1	19	31	13
29	1	18	32	13
30	0	18	32	14
31	0	17	32	15

FIG. 29

Table 30: 4-tap filter for MC interpolation with reference down-sampling at ratio 1.5:1 with 32-phases

Fractional sample position p	4-tap filter for MC interpolation with reference down-sampling at ratio 1.5:1			
	p0	p1	p2	p3
0	13	38	13	0
1	13	37	14	0
2	12	37	15	0
3	11	37	16	0
4	10	37	17	0
5	9	36	19	0
6	8	36	20	0
7	7	36	21	0
8	7	35	22	0
9	6	35	23	0
10	5	34	24	1
11	5	34	25	0
12	4	33	26	1
13	3	32	27	2
14	3	31	29	1
15	2	30	29	3
16	3	29	29	3
17	3	29	30	2
18	1	29	31	3
19	2	27	32	3
20	1	26	33	4
21	0	25	34	5
22	1	24	34	5
23	0	23	35	6
24	0	22	35	7
25	0	21	36	7
26	0	20	36	8
27	0	19	36	9
28	0	17	37	10
29	0	16	37	11
30	0	15	37	12
31	0	14	37	13

FIG. 30

Table 31: 8-tap filter for MC interpolation with reference down-sampling at ratio 2:1

Fractional sample position p	8-tap filter for MC interpolation with reference downsampling at ratio 2:1							
	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

FIG. 31

Table 32: 8-tap filter for MC interpolation with reference down-sampling at ratio 1.5:1

Fractional sample position p	8-tap filter for MC interpolation with reference downsampling at ratio 1.5:1							
	p0	p1	p2	p3	p4	p5	p6	p7
0	-1	-6	20	38	20	-6	-1	0
1	1	-7	18	37	22	-5	-4	2
2	1	-7	16	37	23	-4	-4	2
3	1	-7	14	36	26	-3	-5	2
4	1	-7	12	36	27	-2	-5	2
5	2	-7	11	35	28	-1	-6	2
6	2	-7	9	34	30	0	-6	2
7	2	-7	7	33	31	2	-6	2
8	2	-7	5	32	32	5	-7	2
9	2	-6	2	31	33	7	-7	2
10	2	-6	0	30	34	9	-7	2
11	2	-6	-1	28	35	11	-7	2
12	2	-5	-2	27	36	12	-7	1
13	2	-5	-3	26	36	14	-7	1
14	2	-4	-4	23	37	16	-7	1
15	2	-4	-5	22	37	18	-7	1

FIG. 32

Table 33: 6-tap filter for MC interpolation with reference down-sampling at ratio 2:1 with 16 phases

Fractional sample position p	6-tap filter coefficients for luma 4x4 block MC interpolation with reference downsampling with ratio 2:1					
	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

FIG. 33

Table 34: 6-tap filter for MC interpolation with reference down-sampling at ratio 1.5:1 with 16 phases

Fractional sample position p	6-tap filter coefficients for luma 4x4 block MC interpolation with reference downsampling with ratio 1.5:1					
	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

FIG. 34

Table 35: 4-tap filter for MC interpolation with reference downsampling at ratio 2:1 with 32-phases

Fractional sample position p	4-tap filter for MC interpolation with reference downsampling at ratio 2:1			
	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

FIG. 35

Table 36: 4-tap filter for MC interpolation with reference downsampling at ratio 1.5:1 with 32-phases

Fractional sample position p	4-tap filter for MC interpolation with reference downsampling at ratio 1.5:1			
	p0	p1	p2	p3
0	14	36	14	0
1	14	36	16	-2
2	13	36	17	-2
3	12	36	18	-2
4	12	35	19	-2
5	11	35	20	-2
6	10	34	22	-2
7	9	34	23	-2
8	8	34	24	-2
9	7	34	25	-2
10	7	33	26	-2
11	6	34	26	-2
12	6	34	26	-2
13	5	33	27	-1
14	4	32	28	0
15	4	32	28	0
16	3	29	29	3
17	0	28	32	4
18	0	28	32	4
19	-1	27	33	5
20	-2	26	34	6
21	-2	26	34	6
22	-2	26	33	7
23	-2	25	34	7
24	-2	24	34	8
25	-2	23	34	9
26	-2	22	34	10
27	-2	20	35	11
28	-2	19	35	12
29	-2	18	36	12
30	-2	17	36	13
31	-2	16	36	14

FIG. 36

Table 37: An example of filter coefficient signaling

filter_coefficients( ) {	**Descriptor**
filter_norm	ue(v)
number_of_taps_minus1	ue(v)
number_of_phases_minus1	ue(v)
for (j = 0; j <= number_of_phases_minus1; j += 1) {	
for (i = 0; i <= number_of_taps_minus1; i += 1) {	
coeficient [j] [i]	ue(v)
}	
}	
}	

FIG. 37

Table 38: An example of syntax structure to signal the resampling ratio and corresponding filter set

rpr_filters( ) {	**Descriptor**
number_of_filter_sets	ue(v)
for (i = 0; i <= number_of_filter_sets; i += 1) {	
resample_ratio_denominator	ue(v)
resample_ratio_numerator	ue(v)
filter_coefficients()	
}	
}	

FIG. 38

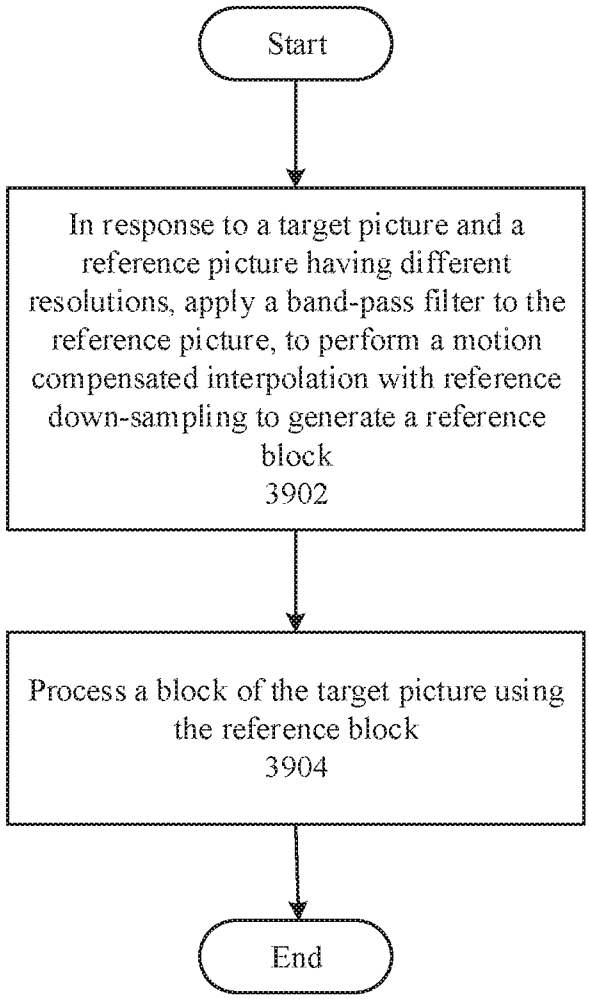

Start
In response to a target picture and a reference picture having different resolutions, apply a band-pass filter to the reference picture, to perform a motion compensated interpolation with reference down-sampling to generate a reference block
3902
Process a block of the target picture using the reference block
3904
End
3900

FIG. 39

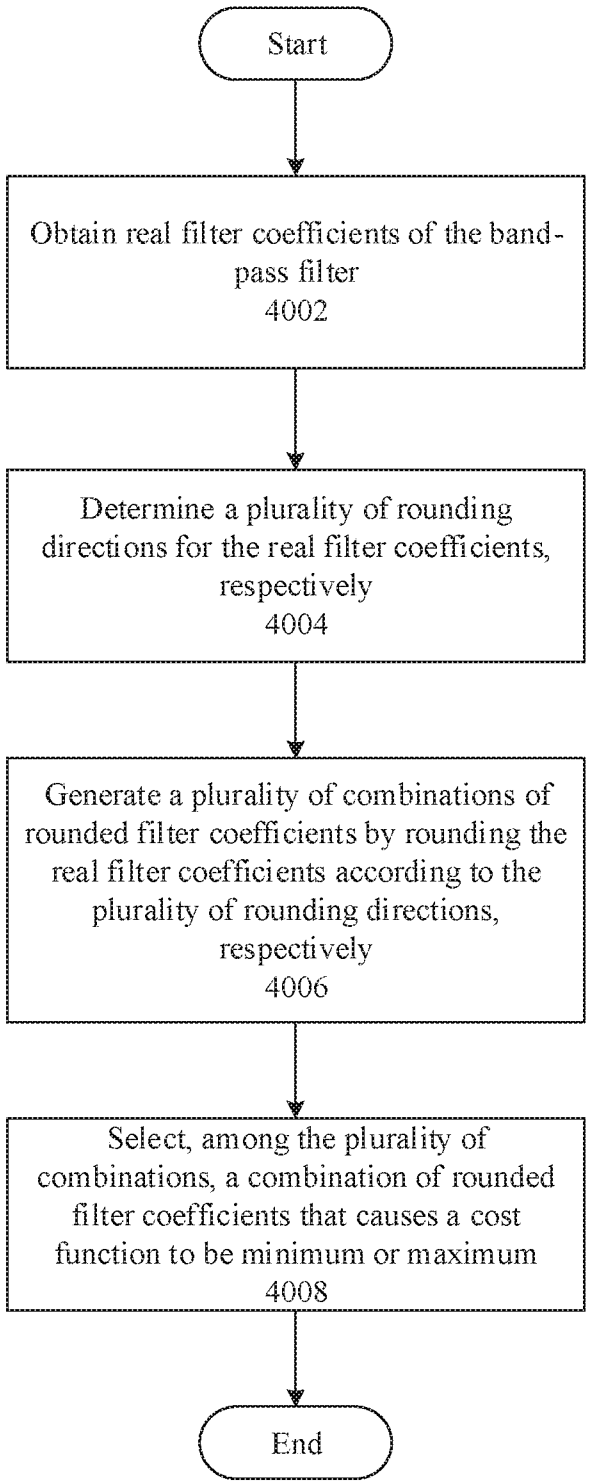

Start
Obtain real filter coefficients of the band-pass filter
4002
Determine a plurality of rounding directions for the real filter coefficients, respectively
4004
Generate a plurality of combinations of rounded filter coefficients by rounding the real filter coefficients according to the plurality of rounding directions, respectively
4006
Select, among the plurality of combinations, a combination of rounded filter coefficients that causes a cost function to be minimum or maximum
4008
End
4000

FIG. 40

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US20/47249

A. CLASSIFICATION OF SUBJECT MATTER

IPC - H04N 19/59, 19/80, 19/117, 19/176, 19/517, 19/51, 19/42, 19/44 (2020.01)

CPC - H04N 19/59, 19/80, 19/117, 19/176, 19/517, 19/51, 19/42, 19/44

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)
See Search History document

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

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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y -- A	US 9,319,729 B2 (MICROSOFT CORPORATION) 19 April 2016 (19.04.2016); abstract, figure 2, column 6, line 54-column 7, line 42, column 9, line 15-column 10, line 61, column 12, lines 51-62, column 15, lines 55-59, claims 1, 10	1, 8, 11, 18, 20 -- 2-7, 9, 10, 12-17, 19
Y -- A	US 8,270,480 B2 (YIN, P et al.) 18 September 2012 (18.09.2012); claim 14	1, 8, 11, 18, 20 -- 2-7, 9, 10, 12-17, 19
Y	WO 2017/158236 A2 (NOKIA TECHNOLOGIES OY) 21 September 2017 (21.09.2017); column 16, lines 6-9, column 17, lines 20-23, column 19, lines 15-27	8, 18
A	US 7,715,573 B1 (YONEMOTO, A et al.) 11 May 2010 (11.05.2010); the entire document	1-20
A	US 2003/0033338 A1 (LINDGREN, U et al.) 13 February 2003 (13.02.2003); the entire document	1-20
A	US 9,179,167 B2 (SAMSUNG ELECTRONICS CO. LTD.) 03 November 2015 (03.11.2015); the entire document	1-20
A	US 9,686,547 B2 (SONY CORPORATION) 20 June 2017 (20.06.2017); the entire document	1-20
A	US 2007/0263714 A1 (BOIS, K et al.) 15 November 2007 (15.11.2007); the entire document	1-20
A	US 7,535,961 B2 (CHO, D et al.) 19 May 2009 (19.05.2009); the entire document	1-20

☐ 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
22 October 2020 (22.10.2020)	23 NOV 2020
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-8300	Authorized officer Shane Thomas Telephone No. PCT Helpdesk: 571-272-4300