



(51) International Patent Classification:
H04N 19/513 (2014.01)

(21) International Application Number:
PCT/CN2020/129887

(22) International Filing Date:
18 November 2020 (18.11.2020)

(25) Filing Language: English

(26) Publication Language: English

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

(72) Inventors: **ZHAO, Lei**; No.969 West Wen Yi Road, Yu
Hang District, Hangzhou, Zhejiang 311121 (CN). **WANG,
Shiqi**; No.969 West Wen Yi Road, Yu Hang District,
Hangzhou, Zhejiang 311121 (CN). **ZHANG, Xinfeng**;
No.969 West Wen Yi Road, Yu Hang District, Hangzhou,
Zhejiang 311121 (CN). **WANG, Shanshe**; No.969 West
Wen Yi Road, Yu Hang District, Hangzhou, Zhejiang
311121 (CN). **MA, Siwei**; No.969 West Wen Yi Road, Yu
Hang District, Hangzhou, Zhejiang 311121 (CN).

(74) Agent: **BEIJING TSINGYUANHUI INTELLECTUAL
PROPERTY LAW FIRM**; Room 101, 1st Floor, Build-

ing 1, No. C-18, Zhichun Road, Haidian District, Beijing
100190 (CN).

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

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

Published:

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

(54) Title: METHODS AND SYSTEMS OF GENERATING VIRTUAL REFERENCE PICTURE FOR VIDEO PROCESSING

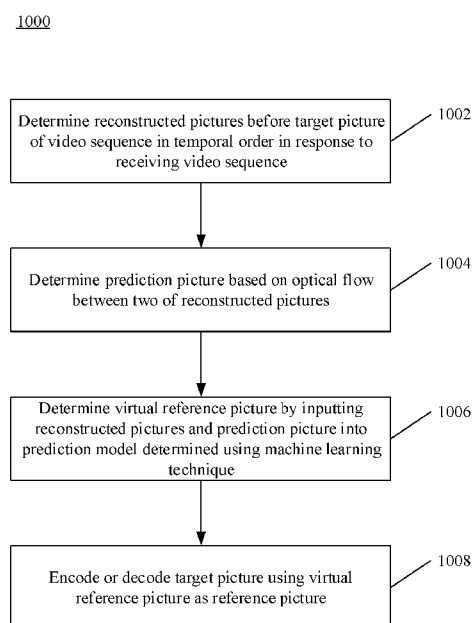


FIG. 10

(57) Abstract: Methods and apparatuses for video processing include: in response to receiving a video sequence, determining a plurality of reconstructed pictures before a target picture of the video sequence in a temporal order; determining a prediction picture based on an optical flow between two of the plurality of reconstructed pictures; determining a virtual reference picture by inputting the plurality of reconstructed pictures and the prediction picture into a prediction model determined using a machine learning technique; and encoding or decoding the target picture using the virtual reference picture as a reference picture.

METHODS AND SYSTEMS OF GENERATING VIRTUAL REFERENCE PICTURE FOR VIDEO PROCESSING

TECHNICAL FIELD

[0001] The present disclosure generally relates to video processing, and more particularly, to methods and systems of generating a virtual reference picture for video processing.

BACKGROUND

[0002] 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.

[0003] Static camera applications, such as in video conferencing and security surveillance, have special demands in video compression for efficient compression methodologies in considerations of characteristics of static camera video systems. Strong background stability is one of the most fundamental characteristics that differentiates static-camera videos from generic videos. The strong background stability incurs high redundancy in static-camera videos, which may limit coding performance of the static-camera videos.

SUMMARY OF THE DISCLOSURE

[0004] The embodiments of present disclosure provide methods and apparatuses for video processing. In an aspect, a non-transitory computer-readable medium is provided. The non-transitory computer-readable medium stores a set of instructions that is executable by at least one processor of an apparatus to cause the apparatus to perform a method. The method includes: in response to receiving a video sequence, determining a plurality of reconstructed

pictures before a target picture of the video sequence in a temporal order; determining a prediction picture based on an optical flow between two of the plurality of reconstructed pictures; determining a virtual reference picture by inputting the plurality of reconstructed pictures and the prediction picture into a prediction model determined using a machine learning technique; and encoding or decoding the target picture using the virtual reference picture as a reference picture.

[0005] In yet another aspect, an apparatus is provided. The apparatus includes a memory configured to store a set of instructions and one or more processors communicatively coupled to the memory and configured to execute the set of instructions to cause the apparatus to perform: in response to receiving a video sequence, determining a plurality of reconstructed pictures before a target picture of the video sequence in a temporal order; determining a prediction picture based on an optical flow between two of the plurality of reconstructed pictures; determining a virtual reference picture by inputting the plurality of reconstructed pictures and the prediction picture into a prediction model determined using a machine learning technique; and encoding or decoding the target picture using the virtual reference picture as a reference picture.

[0006] In yet another aspect, a method is provided. The method includes: in response to receiving a video sequence, determining a plurality of reconstructed pictures before a target picture of the video sequence in a temporal order; determining a prediction picture based on an optical flow between two of the plurality of reconstructed pictures; determining a virtual reference picture by inputting the plurality of reconstructed pictures and the prediction picture into a prediction model determined using a machine learning technique; and encoding or decoding the target picture using the virtual reference picture as a reference picture.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] Embodiments and various aspects of present disclosure are illustrated in the following detailed description and the accompanying figures. Various features shown in the figures are not drawn to scale.

[0008] **Fig. 1** is a schematic diagram illustrating structures of an example video sequence, consistent with some embodiments of this disclosure.

[0009] **Fig. 2A** illustrates a schematic diagram of an example encoding process of a hybrid video coding system, consistent with embodiments of the disclosure.

[0010] **Fig. 2B** illustrates a schematic diagram of another example encoding process of a hybrid video coding system, consistent with embodiments of the disclosure.

[0011] **Fig. 3A** illustrates a schematic diagram of an example decoding process of a hybrid video coding system, consistent with embodiments of the disclosure.

[0012] **Fig. 3B** illustrates a schematic diagram of another example decoding process of a hybrid video coding system, consistent with embodiments of the disclosure.

[0013] **Fig. 4** illustrates a block diagram of an example apparatus for encoding or decoding a video, consistent with some embodiments of this disclosure.

[0014] **Fig. 5** is a schematic representation of a neural network, consistent with some embodiments of this disclosure.

[0015] **Fig. 6** is a flowchart of an example process for determining a prediction picture based on an optical flow, consistent with some embodiments of this disclosure.

[0016] **Fig. 7** is a flowchart of an example process for determining a virtual reference picture, consistent with some embodiments of this disclosure.

[0017] **Fig. 8** is a flowchart of another example process for determining a virtual reference picture, consistent with some embodiments of this disclosure.

[0018] **Fig. 9** is a schematic representation of training performance of a prediction model, consistent with some embodiments of this disclosure.

[0019] **Fig. 10** illustrates a flowchart of an example process for generating a virtual reference picture for video processing, according to some embodiments of this disclosure.

DETAILED DESCRIPTION

[0020] Reference can now be made in detail to example embodiments, examples of which are illustrated in the accompanying drawings. The following description refers to the accompanying drawings in which the same numbers in different drawings represent the same or similar elements unless otherwise represented. The implementations set forth in the following description of example embodiments do not represent all implementations consistent with the 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 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.

[0021] 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 surveillance, conferencing, or live broadcasting.

[0022] 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.”

[0023] 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 trade-off to reduce the needed storage space and the transmission bandwidth.

[0024] The useful information of a picture being encoded (referred to as a “current picture” or “target 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 target picture.

[0025] 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.”

[0026] **Fig. 1** illustrates structures of an example video sequence 100, consistent with some embodiments of this 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.

[0027] 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 the present disclosure does not limit embodiments of the reference pictures as the examples shown in **Fig. 1**.

[0028] 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 the present 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.

[0029] 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.

[0030] Video coding has multiple stages of operations, examples of which are shown in **Figs. 2A-2B** and **Figs. 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 the present 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.

[0031] For example, at a mode decision stage (an example of which is shown 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.

[0032] For another example, at a prediction stage (an example of which is shown in **Figs. 2A-2B**), 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.

[0033] For another example, at a transform stage (an example of which is shown in **Figs. 2A-2B**), 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.

[0034] 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.

[0035] 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.

[0036] 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 the present disclosure does not limit embodiments thereof.

[0037] **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.

[0038] 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.

[0039] 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.

[0040] 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.

[0041] 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.

[0042] 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.

[0043] 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.

[0044] 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.

[0045] 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”).

[0046] 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.

[0047] 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.

[0048] 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.

[0049] 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**.

[0050] **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.

[0051] 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 target 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 target 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.

[0052] 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.

[0053] For another example, at temporal prediction stage 2044, the encoder can perform the inter prediction. For an original BPU of a target 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 target 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 target 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 target 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.

[0054] 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.

[0055] 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 target 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.

[0056] 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 target 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 target 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.

[0057] Still referring to the forward path of process 200B, after spatial prediction stage 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.

[0058] 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 target BPU that has been encoded and reconstructed in the target 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 target picture). If the inter prediction mode has been selected in the forward path, after generating prediction reference 224 (e.g., the target 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.

[0059] **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.

[0060] 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.

[0061] 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.

[0062] 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.

[0063] **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.

[0064] In process 300B, for an encoded basic processing unit (referred to as a “current BPU” or “target BPU”) of an encoded picture (referred to as a “current picture” or “target 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 target BPU by the encoder. For example, if intra prediction was used by the encoder to encode the target 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 target 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 target 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.

[0065] 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**.

[0066] 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 target BPU is decoded using the intra prediction at spatial prediction stage 2042, after generating prediction reference 224

(e.g., the decoded target 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 target picture). If the target 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 target BPU, prediction data can further include parameters of the loop filter (e.g., a loop filter strength).

[0067] **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.

[0068] 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.

[0069] 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.

[0070] 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.

[0071] 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.

[0072] 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 communicatively coupled to a video archive), or the like.

[0073] 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).

[0074] In the quantization and inverse quantization functional blocks (e.g., quantization 214 and inverse quantization 218 of **Fig. 2A** or **Fig. 2B**, inverse quantization 218 of **Fig. 3A**

or **Fig. 3B**), a quantization parameter (QP) is used to determine the amount of quantization (and inverse quantization) applied to the prediction residuals. Initial QP values used for coding of a picture or slice can be signaled at the high level, for example, using a syntax element (e.g., `init_qp_minus26`) in the Picture Parameter Set (PPS) or using a syntax element (e.g., `slice_qp_delta`) in the slice header. Further, the QP values can be adapted at the local level for each CU using delta QP values sent at the granularity of quantization groups.

[0075] Typically, a video codec can use a single reconstructed picture as a reference picture for inter prediction of a target picture. The similarity between the reference picture and the target picture determines the coding efficiency. The higher such similarity is, the higher the coding efficiency can be. The similarity between the single reconstructed picture and the target picture can be low in some scenarios, such as in video clips showing fast motion in the foreground, the background, or both. In such scenarios, a synthesized reference frame (SRF) that synthesizes motion information from multiple pictures can be used for the inter prediction. The SRF can provide higher similarity to the target picture and increase the coding efficiency.

[0076] Using frame interpolation for generating the multiple candidate reference pictures, a virtual reference frame (VRF) can be generated as one type of the SRF and can reduce transmission bandwidth needed for transmitting a video bitstream encoded using the VRF. In some cases, the VRF can be organized at a video-clip level (e.g., in a reference picture buffer) for managing the VRF in a more economical manner. To facilitate coding static-background videos (e.g., videos captured using a static camera) that have strong background stability, a virtual background reference frame (BRF) can be used to model the background of a static-background video. The BRF can reduce background redundancy in the inter prediction for static-background video coding, which can in turn increase the video coding efficiency. In some cases, the BRF can be generated using deep learning techniques.

[0077] Static-background videos also have foreground redundancy. For example, motion of objects in the foreground of a static-background video can conform to one or more patterns. Reducing such foreground redundancy can further improve the video coding efficiency for static-background videos. However, existing video coding solutions have not focused on reducing such foreground redundancy.

[0078] Aspects of this disclosure can provide a technical solution to the challenging problem of reducing foreground redundancy in video coding for static-background or non-static-background videos and can relate to generating a virtual reference picture for video processing, including systems, apparatuses, methods, and non-transitory computer-readable media. For ease of description, some examples are described below with reference to methods, systems, devices, or computer-readable media, with the understanding that discussions of each apply equally to the others. For example, some aspects of methods may be implemented by a computing device or software running thereon. The computing device may include at least one processor as described herein (e.g., a CPU, GPU, DSP, FPGA, ASIC, or any circuitry for performing logical operations on input data) to perform the example methods. Other aspects of such methods may be implemented over a network (e.g., a wired network, a wireless network, or both).

[0079] Aspects of this disclosure relate to using a prediction model determined using a machine learning technique (e.g., a neural network) to predict the motion of foreground objects in a video. For a target picture, the prediction model can generate a VRF that reduces the foreground redundancy, referred to as a foreground reference frame (FRF) herein, based on reconstructed pictures before the target picture in a temporal order and optical flow information determined from the reconstructed pictures. In some embodiments, the prediction model can generate the FRF additionally based on a BRF. The prediction model can be trained using observations of a moving object. In actual application, the prediction

model can predict the motion of a moving foreground object based on its previous moving patterns. Using such a prediction model can reduce the foreground redundancy in video coding, especially for the static-background videos, and thus can further increase the coding efficiency.

[0080] Consistent with some embodiments of this disclosure, a method for generating a virtual reference picture for video processing can include determining multiple reconstructed pictures in a video coding or decoding process, determining a prediction picture based on an optical flow between two of the reconstructed pictures, determining a virtual reference picture (e.g., the FRF) by inputting the reconstructed pictures and the prediction picture into the prediction model, and encoding or decoding a target picture using the virtual reference picture as a reference picture. For static-background videos, the disclosed methods and systems can determine the virtual reference picture by inputting the reconstructed pictures, the prediction picture, and a synthesized background picture (e.g., a BRF) into the prediction model. By doing so, the disclosed methods and systems can predict a motion status for a foreground object based on the reconstructed pictures and can reduce the foreground redundancy and improve the video coding efficiency.

[0081] A neural network, as used herein, can refer to a computing model for analyzing underlying relationships in a set of input data by way of mimicking human brains. Similar to a biological neural network, the neural network can include a set of connected units or nodes (referred to as “neurons”), structured as different layers, where each connection (also referred to as an “edge”) can receive and send a signal between neurons of neighboring layers in a way similar to a synapse in a biological brain. The signal can be any type of data (e.g., a real number). Each neuron can receive one or more signals as an input and output another signal by applying a non-linear function to the inputted signals. Neurons and edges can typically be weighted by corresponding weights to represent the knowledge the neural network has

acquired. During a training process (similar to a learning process of a biological brain), the weights can be adjusted (e.g., by increasing or decreasing their values) to change the strengths of the signals between the neurons to improve the performance accuracy of the neural network. Neurons can apply a thresholding function (referred to as an “activation function”) to its output values of the non-linear function such that a signal is outputted only when an aggregated value (e.g., a weighted sum) of the output values of the non-linear function exceeds a threshold determined by the thresholding function. Different layers of neurons can transform their input signals in different manners (e.g., by applying different non-linear functions or activation functions). The output of the last layer (referred to as an “output layer”) can output the analysis result of the neural network, such as, for example, a categorization of the set of input data (e.g., as in image recognition cases), a numerical result, or any type of output data for obtaining an analytical result from the input data.

[0082] Training of the neural network, as used herein, can refer to a process of improving the accuracy of the output of the neural network. Typically, the training can be categorized into three types: supervised training, unsupervised training, and reinforcement training. In the supervised training, a set of target output data (also referred to as “labels” or “ground truth”) can be generated based on a set of input data using a method other than the neural network. The neural network can then be fed with the set of input data to generate a set of output data that is typically different from the target output data. Based on the difference between the output data and the target output data, the weights of the neural network can be adjusted in accordance with a rule. If such adjustments are successful, the neural network can generate another set of output data more similar to the target output data in a next iteration using the same input data. If such adjustments are not successful, the weights of the neural network can be adjusted again. After a sufficient number of iterations, the training process can be terminated in accordance with one or more predetermined criteria (e.g., the

difference between the final output data and the target output data is below a predetermined threshold, or the number of iterations reaches a predetermined threshold). The trained neural network can be applied to analyze other input data.

[0083] In the unsupervised training, the neural network is trained without any external gauge (e.g., labels) to identify patterns in the input data rather than generating labels for them. Typically, the neural network can analyze shared attributes (e.g., similarities and differences) and relationships among the elements of the input data in accordance with one or more predetermined rules or algorithms (e.g., principal component analysis, clustering, anomaly detection, or latent variable identification). The trained neural network can extrapolate the identified relationships to other input data.

[0084] In the reinforcement learning, the neural network is trained without any external gauge (e.g., labels) in a trial-and-error manner to maximize benefits in decision making. The input data sets of the neural network can be different in the reinforcement training. For example, a reward value or a penalty value can be determined for the output of the neural network in accordance with one or more rules during training, and the weights of the neural network can be adjusted to maximize the reward values (or to minimize the penalty values). The trained neural network can apply its learned decision-making knowledge to other input data.

[0085] It should be noted that the apparatus, systems and methods disclosed herein can be used in various neural network-based architectures, such as deep neural networks (DNNs), convolutional neural networks (CNNs), recurrent neural networks (RNNs), or any architecture or algorithm that can cluster or label input data using machine perceptions (“artificial neurons” or “neurons”). The neural network-based architectures can be used for various applications, such as image classification, three-dimensional object recognition, machine translation, or transductive learning on graphs.

[0086] It should also be noted that the apparatus, systems and methods disclosed herein can also be configured for various processing architectures, such as a central processing unit (CPU), a graphics processing unit (GPU), a neural network processing unit (NPU), a field programmable gate array (FPGA), a tensor processing unit (TPU), a heterogeneous acceleration processing unit (HAPU), an application-specific integrated circuit (ASIC), or any one or more circuits that are capable of processing data.

[0087] By way of example, **Fig. 5** is a schematic representation of a neural network 500. As depicted in **Fig. 5**, neural network 500 can include an input layer 520 that receives inputs, including input 510-1, . . . , input 510- m (m being an integer). An input of neural network 500 in this disclosure can include an image, text, or any other structure or unstructured data for processing by neural network 500. In some embodiments, neural network 500 can receive a plurality of inputs simultaneously. For example, in **Fig. 5**, neural network 500 can receive m inputs simultaneously. In some embodiments, input layer 520 can receive m inputs in succession such that input layer 520 receives input 510-1 in a first cycle (e.g., in a first inference) and pushes data from input 510-1 to a hidden layer (e.g., hidden layer 530-1), then receives a second input in a second cycle (e.g., in a second inference) and pushes data from input the second input to the hidden layer, and so on. Input layer 520 can receive any number of inputs in the simultaneous manner, the successive manner, or any manner of grouping the inputs.

[0088] Input layer 520 can include one or more nodes, including node 520-1, node 520-2, . . . , node 520- a (a being an integer). A node (also referred to as a “machine perception” or a “neuron”) can model the functioning of a biological neuron. Each node can apply an activation function to received inputs (e.g., one or more of input 510-1, . . . , input 510- m). An activation function can include a Heaviside step function, a Gaussian function, a multiquadratic function, an inverse multiquadratic function, a sigmoidal function, a rectified

linear unit (ReLU) function (e.g., a ReLU6 function or a Leaky ReLU function), a hyperbolic tangent (“tanh”) function, or any non-linear function. The output of the activation function can be weighted by a weight associated with the node. A weight can include a positive value between 0 and 1, or any numerical value that can scale outputs of some nodes in a layer more or less than outputs of other nodes in the same layer.

[0089] As further depicted in **Fig. 5**, neural network 500 includes multiple hidden layers, including hidden layer 530-1, . . . , hidden layer 530- n (n being an integer). When neural network 500 includes more than one hidden layer, it can be referred to as a “deep neural network” (DNN). Each hidden layer can include one or more nodes. For example, in **Fig. 5**, hidden layer 530-1 includes node 530-1-1, node 530-1-2, node 530-1-3, . . . , node 530-1- b (b being an integer), and hidden layer 530- n includes node 530- n -1, node 530- n -2, node 530- n -3, . . . , node 530- n - c (c being an integer). Similar to nodes of input layer 520, nodes of the hidden layers can apply the same or different activation functions to outputs from connected nodes of a previous layer, and weight the outputs from the activation functions by weights associated with the nodes.

[0090] As further depicted in **Fig. 5**, neural network 500 can include an output layer 540 that finalizes outputs, including output 550-1, output 550-2, . . . , output 550- d (d being an integer). Output layer 540 can include one or more nodes, including node 540-1, node 540-2, . . . , node 540- d . Similar to nodes of input layer 520 and of the hidden layers, nodes of output layer 540 can apply activation functions to outputs from connected nodes of a previous layer and weight the outputs from the activation functions by weights associated with the nodes.

[0091] Although nodes of each hidden layer of neural network 500 are depicted in **Fig. 5** to be connected to each node of its previous layer and next layer (referred to as “fully connected”), the layers of neural network 500 can use any connection scheme. For example,

one or more layers (e.g., input layer 520, hidden layer 530-1, . . . , hidden layer 530-*n*, or output layer 540) of neural network 500 can be connected using a convolutional scheme, a sparsely connected scheme, or any connection scheme that uses fewer connections between one layer and a previous layer than the fully connected scheme as depicted in **Fig. 5**.

[0092] Moreover, although the inputs and outputs of the layers of neural network 500 are depicted as propagating in a forward direction (e.g., being fed from input layer 520 to output layer 540, referred to as a “feedforward network”) in **Fig. 5**, neural network 500 can additionally or alternatively use backpropagation (e.g., feeding data from output layer 540 towards input layer 520) for other purposes. For example, the backpropagation can be implemented by using long short-term memory nodes (LSTM). Accordingly, although neural network 500 is depicted similar to a convolutional neural network (CNN), neural network 500 can include a recurrent neural network (RNN) or any other neural network.

[0093] Consistent with some embodiments of this disclosure, a processor of a system for generating a virtual reference picture for video processing can perform operations including determining a plurality of reconstructed pictures before a target picture of a video sequence in a temporal order in response to receiving the video sequence. A video sequence in this disclosure can refer to an uncompressed sequence of pictures that can be displayed for viewing. The uncompressed sequence of pictures can be a sequence of original pictures that are not compressed, or a sequence of pictures that are decompressed from a compressed video stream. A target picture (also referred to as a “current picture”) of a video sequence, as used herein, can refer to a picture that is being processed (e.g., being encoded or decoded) by the processor. A reconstructed picture of a video sequence in this disclosure can refer to a picture reconstructed during an encoding process (e.g., in a reconstruction path of the encoding process) or a decoding process. The reconstructed picture can be reconstructed using either inter prediction or intra prediction. The reconstructed picture that is

reconstructed using the inter prediction can use any number of any type of reference pictures, such as a single reference picture or a synthesized reference frame (SRF). A temporal order of pictures in a video sequence, as used herein, can refer to a forward direction of a timeline, in which the pictures are arranged from early to late along the timeline.

[0094] In some embodiments, the plurality of reconstructed pictures can be consecutive in the temporal order. For example, if the plurality of reconstructed pictures includes S (S being an integer greater than one) reconstructed pictures, then the consecutive reconstructed pictures can be represented as $P_{REC} = \{P_1, P_2, \dots, P_S\}$.

[0095] Consistent with some embodiments of this disclosure, the processor of the system can perform operations including determining a prediction picture based on an optical flow between two of the plurality of reconstructed pictures. The prediction picture determined based on the optical flow can be represented as P_{OP} . An optical flow between two pictures, as used herein, can refer to a distribution of velocities of movement of a brightness pattern in an image. Such a brightness pattern can represent a motion of an object, a surface, or an edge in a visual scene caused by a relative motion between an observer and the visual scene. The optical flow can assume a brightness constancy (referred to as a “brightness constancy constraint”) and predict how brightness of each pixel moves across the visual scene over time. The brightness constancy constraint can be a transform of picture coordinates.

[0096] For example, the brightness constancy constraint can be represented by Eq. (1):

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t) \quad \text{Eq. (1)}$$

where $I(x, y, t)$ represents intensity of a pixel at location (x, y) at time t , and

$I(x + \Delta x, y + \Delta y, t + \Delta t)$ represents intensity of a pixel at location $(x + \Delta x, y + \Delta y)$ at time $t + \Delta t$. That is, the constant intensity can represent that the pixel (x, y) “moves” to $(x + \Delta x, y + \Delta y)$ after a time period Δt , which can be referred to as an optical flow

associated with the pixel (x, y) . In such a case, the optical flow can be represented as a vector $(\Delta x, \Delta y)$ pointing from (x, y) to $(x + \Delta x, y + \Delta y)$.

[0097] An optical flow from a first picture to a second picture, as used herein, can refer to an optical flow (e.g., a vector) pointing from a pixel of the first picture to a pixel of the second picture. Optical flows can be determined from a first picture to a second picture for each pixel of the first picture. That is, by moving each pixel of the first picture in accordance with its associated optical flow that points from the first picture to the second picture, the second picture can be obtained.

[0098] The processor can determine the optical flow using various methods, including a phase correlation method, a block-based method, a discrete optimization method, or a differential method (e.g., a Lucas-Kanade method, a Horn-Schunck method, a Buxton-Buxton method, a Black-Jepson method, or a general variational method). For example, the processor can determine the optical flow between the two of the plurality of reconstructed pictures using a FlowNet (e.g., FlowNetS, FlowNetC, FlowNetCSS, or FlowNetSD) algorithm.

[0099] In some embodiments, the two of the plurality of reconstructed pictures for determining the optical flow can be consecutive in the temporal order. For example, the two of the plurality of reconstructed pictures can be any two consecutive ones of $P_{REC} = \{P_1, P_2, \dots, P_S\}$. In some embodiments, the two of the plurality of reconstructed pictures can be immediately before the target picture in the temporal order. For example, assuming the target picture is P_T , P_{REC} can be represented as $\{P_{T-S}, P_{T-S+1}, \dots, P_{T-2}, P_{T-1}\}$, and the two reconstructed pictures for determining the optical flow in P_{REC} can be $\{P_{T-2}, P_{T-1}\}$.

[0100] In some embodiments, to determine the prediction picture based on the optical flow, the processor of the system can perform operations including determining a first optical flow from a first reconstructed picture to a second reconstructed picture. The two of the

plurality of reconstructed pictures can include the first reconstructed picture and the second reconstructed picture. The first reconstructed picture can be after the second reconstructed picture and before the target picture in the temporal order. For example, the first reconstructed picture, the second reconstructed picture, and the target picture can be represented as P_{T-1} , P_{T-2} , and P_T , respectively. The first optical flow can be represented as $O_{T-1 \rightarrow T-2}$. For example, the processor can determine the first optical flow using an algorithm described in Tak-Wai Hui et al., *Liteflownet: A Lightweight Convolutional Neural Network for Optical Flow Estimation*, PROCEEDINGS IEEE CONF. ON COMPUT. VISION & PATTERN RECOGNITION, 2018, at 8981-89.

[0101] The processor can then perform operations including determining a second optical flow from the first reconstructed picture to the target picture based on the first optical flow. The second optical flow (or referred to as a “future optical flow”) can be represented by $O_{T-1 \rightarrow T}$. In some embodiments, the processor can determine $O_{T-1 \rightarrow T}$ by flipping $O_{T-1 \rightarrow T-2}$ in accordance with Eq. (2):

$$O_{T-1 \rightarrow T} = -O_{T-1 \rightarrow T-2} \quad \text{Eq. (2)}$$

[0102] The processor can then perform operations including determining a third optical flow from the target picture to the first reconstructed picture based on the second optical flow. In some embodiments, the processor can determine the third optical flow by inverting $O_{T-1 \rightarrow T}$. The third optical flow (or referred to as an “inverse optical flow”) can be represented by $O_{T \rightarrow T-1}$. For example, the processor can determine $O_{T \rightarrow T-1}$ by inverting $O_{T-1 \rightarrow T}$ in accordance with Eq. (3):

$$O_{T \rightarrow T-1} = IVS(O_{T-1 \rightarrow T}) \quad \text{Eq. (3)}$$

where $IVS(\cdot)$ represents an inverse operation. For example, the processor can determine $O_{T \rightarrow T-1}$ based on $O_{T-1 \rightarrow T}$ using an algorithm described in Javier Sánchez et al., *Computing Inverse Optical Flow*, 52 PATTERN RECOGNITION LETTERS 32 (2015).

[0103] The processor can then perform operations including determining, using a backward warping technique, the prediction picture based on the first reconstructed picture and the third optical flow. The backward warping technique is an image warping technique using an optical flow. Based on an optical flow and a source image (where the optical flow pointing from), the target image (where the optical flow pointing to) can be determined in two techniques: a forward warping technique and a backward warping technique. The forward warping technique can map each pixel of the source image to its corresponding position in the target image in accordance with the optical flow associated with the pixel. However, all adjacent pixels in the source image are not still adjacent in the target image after the mapping, which can cause gaps in the pixels of the target image. To fill those gaps, the forward warping technique can map one pixel in the source image to several positions in the target image, each position assigned with a weight, and then can normalize each pixel in the target image, the process of which can be termed as “splatting.”

[0104] The backward warping technique can determine the target image based on the optical flow and the source image in a different order of procedures. The backward warping technique can first determine an inverse optical flow by applying an inverse operation to the optical flow, in which the inverse optical flow points from the target image to the source image. Then, the backward warping technique can map each pixel (e.g., a target pixel) of the target image to its corresponding position (e.g., a source pixel) in the source image in accordance with the inverse optical flow associated with the pixel. In principle, once the source pixel is determined, the target pixel can be generated by copying the value (e.g., an RGB value) of the source pixel to the position of the target pixel in the target image. However, all adjacent pixels in the target image are not still adjacent in the source image after the mapping. To preserve all of the information included in all pixels in the source image, the backward warping technique can resample the source image by interpolating a source

pixel with its adjacent pixels. For example, an RGB value of the source pixel can be interpolated with RGB values of pixels adjacent to the source pixel to determine an interpolated source pixel corresponding to the target pixel. Then, the target pixel can be generated by copying the value (e.g., an RGB value) of the interpolated source pixel to the position of the target pixel in the target image. Compared with the forward mapping technique, the backward mapping technique can determine the target image with higher quality.

[0105] By way of example, **Fig. 6** is a flowchart of an example process for determining a prediction picture P_{OP} based on an optical flow, consistent with some embodiments of this disclosure. In **Fig. 6**, a first optical flow $O_{T-1 \rightarrow T-2}$ can be determined from two reconstructed images P_{T-1} and P_{T-2} , in which $O_{T-1 \rightarrow T-2}$ points from P_{T-1} to P_{T-2} . A second optical flow $O_{T-1 \rightarrow T}$ can be determined by flipping $O_{T-1 \rightarrow T-2}$ (e.g., in accordance with Eq. (2)). A third optical flow $O_{T \rightarrow T-1}$ can be determined by inverting $O_{T-1 \rightarrow T}$ (e.g., in accordance with Eq. (3)). Then, the prediction picture P_{OP} can be determined based on the reconstructed picture P_{T-1} and the third optical flow $O_{T \rightarrow T-1}$ using a backward warping technique.

[0106] It should be noted that, besides determining P_{OP} based on P_{T-1} and $O_{T \rightarrow T-1}$ using the backward warping technique, in some embodiments, the processor can also determine P_{OP} based on P_{T-1} and the second optical flow $O_{T-1 \rightarrow T}$. This disclosure does not limit the image warping techniques used for determining the prediction picture based on a reconstructed picture to the example embodiments described herein.

[0107] Consistent with some embodiments of this disclosure, the processor of the system can perform operations including determining a virtual reference picture by inputting the plurality of reconstructed pictures and the prediction picture into a prediction model determined using a machine learning technique. The virtual reference picture can be a reference picture that synthesizes motion information from multiple pictures. For example,

the virtual reference picture can be a type of synthesized reference frame (SRF). The machine learning technique can include any combination of a supervised learning technique, an unsupervised learning technique, or a reinforcement learning technique. The prediction model can include an algorithm that has a set of parameters trained using the machine learning technique. For example, the machine learning technique can include a supervised learning technique, and the prediction model can include a neural network model having a set of parameters (e.g., weights or connections) trained using the supervised learning technique. Inputting data into a prediction model, as used herein, can refer to processes or procedures of feeding the data into the prediction model as its input data.

[0108] In some embodiments, the prediction model can include a generative adversarial network (GAN). For example, the GAN can be a static camera video prediction generative adversarial network (SCVP-GAN). It should be noted that the prediction model can be any algorithm trained using a machine learning technique, and this disclosure does not limit the prediction model to any specific example embodiments as described herein.

[0109] In some embodiments, the processor of the system can train the prediction model using original pictures of a video sequence for training (referred to as a “training video sequence”). The training video sequence can be the same or a different video sequence from the video sequence being processed. For example, the training video sequence can be a different video sequence collected from static cameras (e.g., surveillance cameras in public roads or security cameras in campuses). As an example, the majority of the foreground objects in the training video sequence can be pedestrians and vehicles. In some embodiments, to ensure diversity of training data, the number and density of foreground objects in the training video sequence can be required to exceed a threshold value. In some embodiments, as preprocessing procedures, all pictures of the training video sequence can be resized to the same size and normalized to the same range of pixel values.

[0110] In some embodiments, the processor can perform operations including determining a training prediction picture based on an optical flow between a first picture and a second picture of a training video sequence in response to receiving the training video sequence. The first picture can be after the second picture and before a third picture of the training video sequence in the temporal order. The first picture, the second picture, and the third picture can be original pictures that are neither compressed nor reconstructed. For example, the first picture, the second picture, and the third picture of the training video sequence can be represented as P'_{T-1} , P'_{T-2} , and P'_T , respectively. The optical flow between the first picture and the second picture can point from the first picture to the second picture. For example, the optical flow can be represented as $O'_{T-1 \rightarrow T-2}$. The training prediction picture can be represented as P'_{OP} .

[0111] The processor can perform operations including training the prediction model using the machine learning technique. A plurality of pictures of the training video sequence and the training prediction picture can be inputs of the prediction model, and the third picture can be a label for an output of the prediction model. The plurality of pictures can include the first picture and the second picture. In some embodiments, the plurality of pictures can be original pictures. For example, when the prediction model is a GAN (e.g., SCVP-GAN), training the GAN using original pictures of the training video sequence can better train the generative ability of the GAN. As a comparison, training the GAN using reconstructed pictures of the training video sequence can better train the artifact removal ability of the GAN. For ease of description without generality, the following description uses original pictures as example embodiments, but it should be noted that the following description also equally applies to using reconstructed pictures for the training.

[0112] In some embodiments, the plurality of pictures can be consecutive in the temporal order. For example, if the plurality of pictures includes N (N being an integer greater than

one) pictures, then the consecutive pictures can be represented as $P'_{PRE} = \{P'_1, P'_2, \dots, P'_N\}$. In some embodiments, the first picture, the second picture, and the third picture can be consecutive in the temporal order. For example, assuming the second picture P'_{T-1} is P'_N , P'_{PRE} can be represented as $\{P'_{T-N}, P'_{T-N+1}, \dots, P'_{T-2}, P'_{T-1}\}$.

[0113] The training of the prediction model can be similar to the processes or procedures previously described. For example, if the machine learning technique is a supervised learning technique, in the training process, the prediction model can be fed with P'_{PRE} and P'_{OP} as inputs to generate a training reference picture, the process of which can be similar to determining the virtual reference picture as described above. For example, a flipping optical flow (e.g., $O'_{T-1 \rightarrow T}$) can be determined based on the optical flow (e.g., $O'_{T-1 \rightarrow T-2}$) between the first picture (e.g., P'_{T-1}) and the second picture (e.g., P'_{T-2}), such as in accordance with Eq. (2). An inverse optical flow (e.g., $O'_{T \rightarrow T-1}$) can be determined based on the flipping optical flow (e.g., $O'_{T-1 \rightarrow T}$), such as in accordance with Eq. (3). The training prediction picture can then be determined based on the first picture (e.g., P'_{T-1}) and the inverse optical flow (e.g., $O'_{T \rightarrow T-1}$) based on a backward warping technique.

[0114] After determining the training prediction picture, the processor can determine a difference between the training reference picture and the third picture (that is used as a label for the training). Based on the difference, parameters (e.g., weights) of the prediction model can be adjusted in accordance with a rule. If such adjustments are successful, the prediction model can generate another training reference picture more similar to the third picture in a next iteration using the same inputs. If such adjustments are not successful, the parameters of the prediction model can be adjusted again. After a sufficient number of iterations, the training process can be terminated in accordance with one or more predetermined criteria (e.g., the difference between the final training reference picture and the third picture is below a predetermined threshold, or the number of iterations reaches a predetermined threshold).

The trained prediction model can then be used to determine the virtual reference picture using the reconstructed pictures and the prediction picture as inputs.

[0115] In some embodiments, to train the prediction model, the processor of the system can perform operations including determining, in a color space, a first color component of the output of the prediction model based on first color components of the inputs of the prediction model. For example, the color space can be a YUV color space where Y represents a luma component and UV represents two chrominance components (e.g., blue projection and red projection components). In such a case, the first color component can be the Y component, the U component, or the V component. In another example, the color space can be an RGB or CMYK color space. It should be noted that this disclosure does not limit the implementation of the color space to the example embodiments described herein. In some embodiments, the prediction model can input the first color components (e.g., Y components) of the plurality of pictures and the training prediction picture and output the first color component (e.g., a Y component) of the training reference picture.

[0116] The processor can then perform operations including determining a second color component of the output of the prediction model based on second color components of the inputs of the prediction model. For example, the prediction model can input the second color components (e.g., U or V components, or UV components) of the plurality of pictures and the training prediction picture and output the second color component (e.g., a U or V component, or a UV component) of the training reference picture.

[0117] By way of example, **Fig. 7** is a flowchart of an example process for determining a virtual reference picture (P_{VR}), consistent with some embodiments of this disclosure. **Fig. 7** shows a prediction model 702 that can be trained, such as in the processes or procedures as described above. Prediction model 702 can receive a plurality of reconstructed pictures (P_{REC}) of a video sequence being processed and a prediction picture (P_{OP}) as inputs. The

prediction picture P_{OP} can be determined based on an optical flow between two of P_{REC} (e.g., as illustrated and described in association with **Fig. 6**). Prediction model 702 can then output the virtual reference picture P_{VR} . Because P_{OP} can provide motion information and preliminary prediction of foreground objects by the optical flows, P_{VR} can integrate such motion information and preliminary prediction.

[0118] In some embodiments, if the video sequence is a static-background video, to determine the virtual reference picture, the processor of the system can perform operations including determining a synthesized background picture for the video sequence. The synthesized background picture can be a SRF (e.g., a VRF) that can reduce background redundancy for the static-background video. For example, the processor can determine the synthesized background picture based on statistics of characteristics of a plurality of original or reconstructed pictures of the video sequence. As an example, the processor can use M (e.g., M being 600) pictures of the video sequence to determine statistics of motion information of each pixel of the M pictures, determine the background pixels, and synthesize the background pixels to generate the synthesized background picture. It should be noted that the processor can employ any existing technique for determining the synthesized background picture.

[0119] After determining the synthesized background picture, the processor can further perform operations including determining the virtual reference picture by inputting the plurality of reconstructed pictures, the prediction picture, and the synthesized background picture into the prediction model. That is, the prediction model can receive the synthesized background picture as an additional input for determining the virtual reference picture.

[0120] In some embodiments, if the video sequence is a static-background video, the processor of the system can train the prediction model using a training video sequence as follows. In some embodiments, the processor can perform operations including determining

a training prediction picture (e.g., P'_{OP}) based on an optical flow (e.g., $O'_{T-1 \rightarrow T-2}$) between a first picture (e.g., P'_{T-1}) and a second picture (e.g., P'_{T-2}) of a training video sequence in response to receiving the training video sequence. The first picture can be after the second picture and before a third picture (e.g., P'_T) of the training video sequence in the temporal order.

[0121] The processor can further perform operations including receiving a background picture of the training video sequence, in which the background picture includes no foreground object. The background picture can be represented as P'_{BG} . In some embodiments, the background picture can be predetermined. In some embodiments, the background picture can be automatically determined based on the processor. For example, the processor can perform a foreground object preprocessing technique (e.g., object segmentation or extraction) to pictures of the training video sequence, and determines a picture where no foreground object can be segmented or extracted as the background picture.

[0122] The processor can further perform operations including training the prediction model using the machine learning technique. A plurality of pictures (e.g., P'_{PRE}) of the training video sequence, the training prediction picture (e.g., P'_{OP}), and the background picture (e.g., P'_{BG}) can be inputs of the prediction model, and the third picture (e.g., P'_T) can be a label for an output of the prediction model. The plurality of pictures can include the first picture and the second picture. The training process can be similar to the above-described training process (e.g., using a supervised learning technique) except that the training model receives P'_{BG} as an additional input, and will not be repeated hereinafter for ease of explanation.

[0123] By way of example, **Fig. 8** is a flowchart of another example process for determining a virtual reference picture (P_{VR}), consistent with some embodiments of this disclosure. **Fig. 8** shows a prediction model 802 that can be trained, such as in the processes

or procedures as described above where the training model receives P'_{BG} as an additional input. Prediction model 802 can receive a plurality of reconstructed pictures (P_{REC}) of a video sequence being processed, a prediction picture (P_{OP}), and a synthesized background picture (P_{SB}) as inputs. The prediction picture P_{OP} can be determined based on an optical flow between two of P_{REC} (e.g., as illustrated and described in association with **Fig. 6**). The synthesized background picture P_{SB} can be determined based on P_{REC} . Prediction model 802 can then output the virtual reference picture P_{VR} . Because P_{OP} can provide motion information and preliminary prediction of foreground objects by the optical flows, P_{VR} can integrate such motion information and preliminary prediction. Because P_{SB} can provide image information of exposed background regions, P_{VR} can also integrate such image information.

[0124] By way of example, **Fig. 9** is a schematic representation of training performance of a prediction model, consistent with some embodiments of this disclosure. For example, the prediction model in **Fig. 9** can be prediction model 702 in **Fig. 7** or prediction model 802 in **Fig. 8**. In **Fig. 9**, the prediction model is an SCVP-GAN. **Fig. 9** illustrates curves of peak signal to noise ratio (PSNR) over iterations of the training of the SCVP-GAN. The multiple curves represent that the SCVP-GAN receives multiple different inputs for the training. For example, the curve having a solid circle legend represents a PSNR curve where the SCVP-GAN receives only P'_{PRE} as training input data. The curve having a solid square legend represents a PSNR curve where the SCVP-GAN receives P'_{PRE} and P'_{BG} as training input data. The curve having a hollow triangle legend represents a PSNR curve where the SCVP-GAN receives P'_{PRE} and P'_{OP} as training input data. The curve having a solid pentacle legend represents a PSNR curve where the SCVP-GAN receives P'_{PRE} , P'_{OP} , and P'_{BG} as training input data.

[0125] As illustrated in **Fig. 9**, compared with the baseline case where the SCVP-GAN receives only P'_{PRE} as training input data, the additional training input data P'_{BG} and P'_{OP} can greatly improve training performance of the SCVP-GAN. For example, when the SCVP-GAN receives P'_{PRE} and P'_{BG} as training input data, approximately 0.5 dB PSNR improvement can be achieved after the training (e.g., after 700,000 iterations). As another example, when the SCVP-GAN receives P'_{PRE} and P'_{OP} as training input data, approximately 1.2 dB PSNR improvement can be achieved compared with the baseline case. In another example, when the SCVP-GAN receives P'_{PRE} , P'_{OP} , and P'_{BG} as training input data, approximately 1.4 dB PSNR improvement can be achieved compared with the baseline case.

[0126] Consistent with some embodiments of this disclosure, the processor of the system can perform operations including encoding or decoding the target picture using the virtual reference picture as a reference picture. By way of example, the virtual reference picture can be included in the prediction reference 224 in process 200A, 200B, 300A, or 300B in **Figs. 2A-3B**. Compared with existing solutions, using the virtual reference picture as a reference for encoding or decoding the target picture as described herein can reduce foreground redundancy in video coding for any video sequence having generic foreground contents and does not require any preprocessing (e.g., object segmentation or extraction) of the foreground contents for reducing foreground redundancy, and thus can increase coding efficiency. In cases where the video sequence is a static-background video, the virtual reference picture can be determined additionally based on a synthesized background picture and thus can reduce both foreground redundancy and background redundancy for the static-background video. Further, because all inputs for determining the virtual reference picture are ultimately based on reconstructed pictures, if an encoder transmits an encoded video bitstream over a network to a decoder, the decoder can reconstruct the virtual reference picture based only on the transmitted data and rely on no additional data.

[0127] Consistent with some embodiments of this disclosure, the processor of the system can further perform operations including storing the virtual reference picture into a reference picture set for encoding or decoding the video sequence. For example, the processor can use the virtual reference picture as a long-term reference picture for encoding or decoding one or more future pictures in the video sequence. By way of example, the reference picture set can be stored in buffer 234 in process 200B or 200B in **Fig. 2B** or **3B**.

[0128] Consistent with some embodiments of this disclosure, the processor of the system can further perform operations including applying a rate-distortion optimization technique to the virtual reference picture. The rate-distortion optimization technique can be used for maintaining coding quality. For example, the processor apply the rate-distortion optimization technique to determine the usage of virtual reference picture in order to achieve minimal rate-distortion cost.

[0129] Consistent with some embodiments of this disclosure, aspects of this disclosure can be implemented in a video coding standard (e.g., the H.265/HEVC standard). For example, the method of generating a virtual reference picture for video processing can be implemented in an HEVC encoder or decoder as a component for determining prediction reference (e.g., prediction reference 224 in **Figs. 2A-3B**). In some embodiments, in an HEVC Test Model (e.g., HM-16.6) that implements the provided method of generating the virtual reference picture, substantive Bjontegaard rate difference (“BD-rate”) gains can be achieved based on observations of experiments. For example, in a low-delay-B (LDB) configuration, an overall 2.1% luma bitrate savings can be obtained. In another example, in a low-delay-P (LDP) configuration, an overall 2.6% luma bitrate savings can be obtained. The experiment result demonstrates the great improvement provided by the methods and systems as provided herein.

[0130] **Fig. 10** illustrate a flowchart of an example process 1000 for generating a virtual reference picture for video processing, according to some embodiments of this disclosure. Method 1000 can be performed by at least one processor (e.g., processor 402 in **Fig. 4**) associated with a video encoder (e.g., an encoder described in association with **Figs. 2A-2B**) or a video decoder (e.g., a decoder described in association with **Figs. 3A-3B**). In some embodiments, method 1000 can be implemented as a computer program product (e.g., embodied in a computer-readable medium) that includes computer-executable instructions (e.g., program codes) to be executed by a computer (e.g., apparatus 400 in **Fig. 4**). In some embodiments, method 1000 can be implemented as a hardware product (e.g., memory 404 in **Fig. 4**) that stores computer-executable instructions (e.g., program instructions in memory 404 in **Fig. 4**), and the hardware product can be a standalone or integrated part of the computer.

[0131] At step 1002, in response to a processor (e.g., processor 402 in **Fig. 4**) receiving a video sequence (e.g., video sequence 202 in **Figs. 2A-2B**), the processor can determine a plurality of reconstructed pictures (e.g., P_{REC} as described herein) before a target picture (e.g., P_T as described herein) of the video sequence in a temporal order. In some embodiments, the plurality of reconstructed pictures can be consecutive in the temporal order. For example, the plurality of reconstructed pictures can be $\{P_1, P_2, \dots, P_S\}$ as described herein, where S is an integer greater than one.

[0132] At step 1004, the processor can determine a prediction picture (e.g., P_{OP} as described herein) based on an optical flow (e.g., the first optical flow $O_{T-1 \rightarrow T-2}$ as described herein) between two of the plurality of reconstructed pictures. In some embodiments, the two of the plurality of reconstructed pictures can be consecutive in the temporal order. For example, the two of the plurality of reconstructed pictures can be any two consecutive ones of $P_{REC} = \{P_1, P_2, \dots, P_S\}$. In some embodiments, the two of the plurality of reconstructed

pictures can be immediately before the target picture in the temporal order. For example, assuming the target picture is P_T , P_{REC} can be represented as $\{P_{T-S}, P_{T-S+1}, \dots, P_{T-2}, P_{T-1}\}$, and the two reconstructed pictures for determining the optical flow in P_{REC} can be $\{P_{T-2}, P_{T-1}\}$.

[0133] At step 1006, the processor can determine a virtual reference picture (e.g., P_{VR} as described herein) by inputting the plurality of reconstructed pictures and the prediction picture into a prediction model (e.g., prediction model 702 in **Fig. 7**) determined using a machine learning technique (e.g., a supervised learning technique, an unsupervised learning technique, an reinforcement learning technique, or any combination thereof). In some embodiments, the prediction model can include a generative adversarial network (GAN). For example, the GAN can be a static camera video prediction generative adversarial network (SCVP-GAN).

[0134] In some embodiments, to determine the prediction picture based on the optical flow, the processor can determine a first optical flow (e.g., the first optical flow $O_{T-1 \rightarrow T-2}$ as described herein) from a first reconstructed picture (e.g., P_{T-1}) to a second reconstructed picture (e.g., P_{T-2}). The two of the plurality of reconstructed pictures (e.g., $P_{REC} = \{P_{T-S}, P_{T-S+1}, \dots, P_{T-2}, P_{T-1}\}$ as described herein) can include the first reconstructed picture and the second reconstructed picture. The first reconstructed picture can be after the second reconstructed picture and before the target picture (e.g., P_T) in the temporal order. The processor can then determine a second optical flow (e.g., $O_{T-1 \rightarrow T}$ as described in association with Eq. (2)) from the first reconstructed picture to the target picture based on the first optical flow. The processor can then determine a third optical flow (e.g., $O_{T \rightarrow T-1}$ as described in association with Eq. (3)) from the target picture to the first reconstructed picture based on the second optical flow. After that, the processor can then determine, using a backward warping

technique, the prediction picture based on the first reconstructed picture and the third optical flow.

[0135] In some embodiments, the processor can train the prediction model as follows. The processor can determine a training prediction picture (e.g., P'_{OP} as described herein) based on an optical flow (e.g., $O'_{T-1 \rightarrow T-2}$ as described herein) between a first picture (e.g., P'_{T-1} as described herein) and a second picture (e.g., P'_{T-2} as described herein) of a training video sequence in response to receiving the training video sequence. The first picture can be after the second picture and before a third picture (e.g., P'_T as described herein) of the training video sequence in the temporal order. The processor can then train the prediction model using the machine learning technique. A plurality of pictures (e.g., P'_{PRE} as described herein) of the training video sequence and the training prediction picture can be inputs of the prediction model, and the third picture can be a label for an output of the prediction model. The plurality of pictures (e.g., P'_{PRE} as described herein) can include the first picture and the second picture.

[0136] In some embodiments, the plurality of pictures (e.g., $P'_{PRE} = \{P'_{T-N}, P'_{T-N+1}, \dots, P'_{T-2}, P'_{T-1}\}$ as described herein) can be consecutive in the temporal order. In some embodiments, the first picture, the second picture, and the third picture can be consecutive in the temporal order. For example, assuming the second picture P'_{T-1} is P'_N , P'_{PRE} can be represented as $\{P'_{T-N}, P'_{T-N+1}, \dots, P'_{T-2}, P'_{T-1}\}$.

[0137] In some embodiments, to train the prediction model, the processor can determine, in a color space (e.g., YUV, RGB, CMYK, or any color space), a first color component (e.g., a Y component in a YUV color space) of the output of the prediction model based on first color components (e.g., Y components in the YUV color space) of the inputs of the prediction model. The processor can then determine a second color component (e.g., a U, V, or UV component in the YUV color space) of the output of the prediction model based on second

color components (e.g., U, V, or UV components in the YUV color space) of the inputs of the prediction model.

[0138] In some embodiments, to determine the virtual reference picture, the processor of the system can determine a synthesized background picture for the video sequence. The processor can then determine the virtual reference picture by inputting the plurality of reconstructed pictures, the prediction picture, and the synthesized background picture into the prediction model.

[0139] In some embodiments, to train a prediction model that receives the plurality of reconstructed pictures, the prediction picture, and the synthesized background picture as inputs, the processor can determine a training prediction picture (e.g., P'_{OP}) based on an optical flow (e.g., $O'_{T-1 \rightarrow T-2}$) between a first picture (e.g., P'_{T-1}) and a second picture (e.g., P'_{T-2}) of a training video sequence in response to receiving the training video sequence. The first picture can be after the second picture and before a third picture (e.g., P'_T) of the training video sequence in the temporal order. The processor can then receive a background picture (e.g., P'_{BG} as described herein) of the training video sequence, in which the background picture includes no foreground object. The processor can further train the prediction model using the machine learning technique. A plurality of pictures (e.g., P'_{PRE}) of the training video sequence, the training prediction picture (e.g., P'_{OP}), and the background picture (e.g., P'_{BG}) can be inputs of the prediction model, and the third picture (e.g., P'_T) can be a label for an output of the prediction model. The plurality of pictures can include the first picture and the second picture.

[0140] Still referring to **Fig. 10**, at step 1008, the processor can encode or decode the target picture using the virtual reference picture as a reference picture. In some embodiments, the processor can further store the virtual reference picture into a reference picture set (e.g., in buffer 234 in process 200B or 200B in **Fig. 2B** or **3B**) for encoding or

decoding the video sequence. In some embodiments, the processor can further apply a rate-distortion optimization technique to the virtual reference picture.

[0141] In some embodiments, a non-transitory computer-readable storage medium including instructions is also provided, and the instructions can 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 can include one or more processors (CPUs), an input/output interface, a network interface, and/or a memory.

[0142] The embodiments can further be described using the following clauses:

1. A non-transitory computer-readable medium storing a set of instructions that is executable by at least one processor of an apparatus to cause the apparatus to perform a method, the method comprising:

- in response to receiving a video sequence, determining a plurality of reconstructed pictures before a target picture of the video sequence in a temporal order;

- determining a prediction picture based on an optical flow between two of the plurality of reconstructed pictures;

- determining a virtual reference picture by inputting the plurality of reconstructed pictures and the prediction picture into a prediction model determined using a machine learning technique; and

encoding or decoding the target picture using the virtual reference picture as a reference picture.

2. The non-transitory computer-readable medium of clause 1, wherein the plurality of reconstructed pictures are consecutive in the temporal order.

3. The non-transitory computer-readable medium of any of clauses 1-2, wherein the two of the plurality of reconstructed pictures are consecutive in the temporal order.

4. The non-transitory computer-readable medium of clause 3, wherein the two of the plurality of reconstructed pictures are immediately before the target picture in the temporal order.

5. The non-transitory computer-readable medium of any of clauses 1-4, wherein the prediction model comprises a generative adversarial network.

6. The non-transitory computer-readable medium of any of clauses 1-5, wherein determining the prediction picture based on the optical flow between the two of the plurality of reconstructed pictures comprises:

determining a first optical flow from a first reconstructed picture to a second reconstructed picture, wherein the two of the plurality of reconstructed pictures comprises the first reconstructed picture and the second reconstructed picture, and the first reconstructed picture is after the second reconstructed picture and before the target picture in the temporal order;

determining a second optical flow from the first reconstructed picture to the target picture based on the first optical flow;

determining a third optical flow from the target picture to the first reconstructed picture based on the second optical flow; and

determining, using a backward warping technique, the prediction picture based on the first reconstructed picture and the third optical flow.

7. The non-transitory computer-readable medium of any of clauses 1-6, wherein the set of instructions that is executable by the at least one processor of the apparatus causes the apparatus to further perform:

storing the virtual reference picture into a reference picture set for encoding or decoding the video sequence.

8. The non-transitory computer-readable medium of any of clauses 1-7, wherein the set of instructions that is executable by the at least one processor of the apparatus causes the apparatus to further perform:

applying a rate-distortion optimization technique to the virtual reference picture.

9. The non-transitory computer-readable medium of any of clauses 1-8, wherein the set of instructions that is executable by the at least one processor of the apparatus causes the apparatus to further perform:

in response to receiving a training video sequence, determining a training prediction picture based on an optical flow between a first picture and a second picture of the training video sequence, wherein the first picture is after the second picture and before a third picture of the training video sequence in the temporal order; and

training the prediction model using the machine learning technique, wherein a plurality of pictures of the training video sequence and the training prediction picture are inputs of the prediction model, and the third picture is a label for an output of the prediction model, and wherein the plurality of pictures comprises the first picture and the second picture.

10. The non-transitory computer-readable medium of clause 9, wherein the plurality of pictures are consecutive in the temporal order.

11. The non-transitory computer-readable medium of any of clauses 9-10, wherein the first picture, the second picture, and the third picture are consecutive in the temporal order.

12. The non-transitory computer-readable medium of any of clauses 9-11, wherein training the prediction model comprises:

determining, in a color space, a first color component of the output of the prediction model based on first color components of the inputs of the prediction model; and

determining a second color component of the output of the prediction model based on second color components of the inputs of the prediction model.

13. The non-transitory computer-readable medium of any of clauses 1-8, wherein determining the virtual reference picture comprises:

determining a synthesized background picture for the video sequence; and

determining the virtual reference picture by inputting the plurality of reconstructed pictures, the prediction picture, and the synthesized background picture into the prediction model.

14. The non-transitory computer-readable medium of clause 13, wherein the set of instructions that is executable by the at least one processor of the apparatus causes the apparatus to further perform:

in response to receiving a training video sequence, determining a training prediction picture based on an optical flow between a first picture and a second picture of the training video sequence, wherein the first picture is after the second

picture and before a third picture of the training video sequence in the temporal order;

receiving a background picture of the training video sequence, wherein the background picture comprises no foreground object; and

training the prediction model using the machine learning technique, wherein a plurality of pictures of the training video sequence, the training prediction picture, and the background picture are inputs of the prediction model, and the third picture is a label for an output of the prediction model, and wherein the plurality of pictures comprises the first picture and the second picture.

15. An apparatus, comprising:

a memory configured to store a set of instructions; and

one or more processors communicatively coupled to the memory and configured to execute the set of instructions to cause the apparatus to perform:

in response to receiving a video sequence, determining a plurality of reconstructed pictures before a target picture of the video sequence in a temporal order;

determining a prediction picture based on an optical flow between two of the plurality of reconstructed pictures;

determining a virtual reference picture by inputting the plurality of reconstructed pictures and the prediction picture into a prediction model determined using a machine learning technique; and

encoding or decoding the target picture using the virtual reference picture as a reference picture.

16. The apparatus of clause 15, wherein the plurality of reconstructed pictures are consecutive in the temporal order.

17. The apparatus of any of clauses 15-16, wherein the two of the plurality of reconstructed pictures are consecutive in the temporal order.

18. The apparatus of clause 17, wherein the two of the plurality of reconstructed pictures are immediately before the target picture in the temporal order.

19. The apparatus of any of clauses 15-18, wherein the prediction model comprises a generative adversarial network.

20. The apparatus of any of clauses 15-19, wherein determining the prediction picture based on the optical flow between the two of the plurality of reconstructed pictures comprises:

determining a first optical flow from a first reconstructed picture to a second reconstructed picture, wherein the two of the plurality of reconstructed pictures comprises the first reconstructed picture and the second reconstructed picture, and the first reconstructed picture is after the second reconstructed picture and before the target picture in the temporal order;

determining a second optical flow from the first reconstructed picture to the target picture based on the first optical flow;

determining a third optical flow from the target picture to the first reconstructed picture based on the second optical flow; and

determining, using a backward warping technique, the prediction picture based on the first reconstructed picture and the third optical flow.

21. The apparatus of any of clauses 15-20, wherein the one or more processors are further configured to execute the set of instructions to cause the apparatus to perform:

storing the virtual reference picture into a reference picture set for encoding or decoding the video sequence.

22. The apparatus of any of clauses 15-21, wherein the one or more processors are further configured to execute the set of instructions to cause the apparatus to perform:

applying a rate-distortion optimization technique to the virtual reference picture.

23. The apparatus of any of clauses 15-22, wherein the one or more processors are further configured to execute the set of instructions to cause the apparatus to perform:

in response to receiving a training video sequence, determining a training prediction picture based on an optical flow between a first picture and a second picture of the training video sequence, wherein the first picture is after the second picture and before a third picture of the training video sequence in the temporal order; and

training the prediction model using the machine learning technique, wherein a plurality of pictures of the training video sequence and the training prediction picture are inputs of the prediction model, and the third picture is a label for an output of the prediction model, and wherein the plurality of pictures comprises the first picture and the second picture.

24. The apparatus of clause 23, wherein the plurality of pictures are consecutive in the temporal order.

25. The apparatus of any of clauses 23-24, wherein the first picture, the second picture, and the third picture are consecutive in the temporal order.

26. The apparatus of any of clauses 23-25, wherein training the prediction model comprises:

determining, in a color space, a first color component of the output of the prediction model based on first color components of the inputs of the prediction model; and

determining a second color component of the output of the prediction model based on second color components of the inputs of the prediction model.

27. The apparatus of any of clauses 15-22, wherein determining the virtual reference picture comprises:

determining a synthesized background picture for the video sequence; and

determining the virtual reference picture by inputting the plurality of reconstructed pictures, the prediction picture, and the synthesized background picture into the prediction model.

28. The apparatus of clause 27, wherein the one or more processors are further configured to execute the set of instructions to cause the apparatus to perform:

in response to receiving a training video sequence, determining a training prediction picture based on an optical flow between a first picture and a second picture of the training video sequence, wherein the first picture is after the second picture and before a third picture of the training video sequence in the temporal order;

receiving a background picture of the training video sequence, wherein the background picture comprises no foreground object; and

training the prediction model using the machine learning technique, wherein a plurality of pictures of the training video sequence, the training

prediction picture, and the background picture are inputs of the prediction model, and the third picture is a label for an output of the prediction model, and wherein the plurality of pictures comprises the first picture and the second picture.

29. A method, comprising:

in response to receiving a video sequence, determining a plurality of reconstructed pictures before a target picture of the video sequence in a temporal order;

determining a prediction picture based on an optical flow between two of the plurality of reconstructed pictures;

determining a virtual reference picture by inputting the plurality of reconstructed pictures and the prediction picture into a prediction model determined using a machine learning technique; and

encoding or decoding the target picture using the virtual reference picture as a reference picture.

30. The method of clause 29, wherein the plurality of reconstructed pictures are consecutive in the temporal order.

31. The method of any of clauses 29-30, wherein the two of the plurality of reconstructed pictures are consecutive in the temporal order.

32. The method of clause 31, wherein the two of the plurality of reconstructed pictures are immediately before the target picture in the temporal order.

33. The method of any of clauses 29-32, wherein the prediction model comprises a generative adversarial network.

34. The method of any of clauses 29-33, wherein determining the prediction picture based on the optical flow between the two of the plurality of reconstructed pictures comprises:

determining a first optical flow from a first reconstructed picture to a second reconstructed picture, wherein the two of the plurality of reconstructed pictures comprises the first reconstructed picture and the second reconstructed picture, and the first reconstructed picture is after the second reconstructed picture and before the target picture in the temporal order;

determining a second optical flow from the first reconstructed picture to the target picture based on the first optical flow;

determining a third optical flow from the target picture to the first reconstructed picture based on the second optical flow; and

determining, using a backward warping technique, the prediction picture based on the first reconstructed picture and the third optical flow.

35. The method of any of clauses 29-34, further comprising:

storing the virtual reference picture into a reference picture set for encoding or decoding the video sequence.

36. The method of any of clauses 29-35, further comprising:

applying a rate-distortion optimization technique to the virtual reference picture.

37. The method of any of clauses 29-36, further comprising:

in response to receiving a training video sequence, determining a training prediction picture based on an optical flow between a first picture and a second picture of the training video sequence, wherein the first picture is after the second

picture and before a third picture of the training video sequence in the temporal order; and

training the prediction model using the machine learning technique, wherein a plurality of pictures of the training video sequence and the training prediction picture are inputs of the prediction model, and the third picture is a label for an output of the prediction model, and wherein the plurality of pictures comprises the first picture and the second picture.

38. The method of clause 37, wherein the plurality of pictures are consecutive in the temporal order.

39. The method of any of clauses 37-38, wherein the first picture, the second picture, and the third picture are consecutive in the temporal order.

40. The method of any of clauses 37-39, wherein training the prediction model comprises:

determining, in a color space, a first color component of the output of the prediction model based on first color components of the inputs of the prediction model; and

determining a second color component of the output of the prediction model based on second color components of the inputs of the prediction model.

41. The method of any of clauses 29-36, wherein determining the virtual reference picture comprises:

determining a synthesized background picture for the video sequence; and

determining the virtual reference picture by inputting the plurality of reconstructed pictures, the prediction picture, and the synthesized background picture into the prediction model.

42. The method of clause 41, further comprising:

in response to receiving a training video sequence, determining a training prediction picture based on an optical flow between a first picture and a second picture of the training video sequence, wherein the first picture is after the second picture and before a third picture of the training video sequence in the temporal order;

receiving a background picture of the training video sequence, wherein the background picture comprises no foreground object; and

training the prediction model using the machine learning technique, wherein a plurality of pictures of the training video sequence, the training prediction picture, and the background picture are inputs of the prediction model, and the third picture is a label for an output of the prediction model, and wherein the plurality of pictures comprises the first picture and the second picture.

[0143] 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.

[0144] 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 component can include A or B, then, unless specifically stated otherwise or infeasible, the component can include A, or B, or A and B. As a second example, if it is stated that a component can include A, B, or C, then, unless specifically stated otherwise or infeasible, the component can include A, or B, or C, or A and B, or A and C, or B and C, or A and B and C.

[0145] 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 can 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 the present disclosure can be implemented by hardware, or software, or a combination of hardware and software. One of ordinary skill in the art can also understand that multiple ones of the above described modules/units can be combined as one module/unit, and each of the above described modules/units can be further divided into a plurality of sub-modules/sub-units.

[0146] 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 disclosure disclosed herein. It is intended that the specification and examples be considered as example only, with a true scope and spirit of the disclosure 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.

[0147] In the drawings and specification, there have been disclosed example 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 non-transitory computer-readable medium storing a set of instructions that is executable by at least one processor of an apparatus to cause the apparatus to perform a method, the method comprising:
 - in response to receiving a video sequence, determining a plurality of reconstructed pictures before a target picture of the video sequence in a temporal order;
 - determining a prediction picture based on an optical flow between two of the plurality of reconstructed pictures;
 - determining a virtual reference picture by inputting the plurality of reconstructed pictures and the prediction picture into a prediction model determined using a machine learning technique; and
 - encoding or decoding the target picture using the virtual reference picture as a reference picture.
2. The non-transitory computer-readable medium of claim 1, wherein the plurality of reconstructed pictures are consecutive in the temporal order.
3. The non-transitory computer-readable medium of claim 1, wherein the two of the plurality of reconstructed pictures are consecutive in the temporal order.
4. The non-transitory computer-readable medium of claim 3, wherein the two of the plurality of reconstructed pictures are immediately before the target picture in the temporal order.

5. The non-transitory computer-readable medium of claim 1, wherein the prediction model comprises a generative adversarial network.

6. The non-transitory computer-readable medium of claim 1, wherein determining the prediction picture based on the optical flow between the two of the plurality of reconstructed pictures comprises:

determining a first optical flow from a first reconstructed picture to a second reconstructed picture, wherein the two of the plurality of reconstructed pictures comprises the first reconstructed picture and the second reconstructed picture, and the first reconstructed picture is after the second reconstructed picture and before the target picture in the temporal order;

determining a second optical flow from the first reconstructed picture to the target picture based on the first optical flow;

determining a third optical flow from the target picture to the first reconstructed picture based on the second optical flow; and

determining, using a backward warping technique, the prediction picture based on the first reconstructed picture and the third optical flow.

7. The non-transitory computer-readable medium of claim 1, wherein the set of instructions that is executable by the at least one processor of the apparatus causes the apparatus to further perform:

storing the virtual reference picture into a reference picture set for encoding or decoding the video sequence.

8. The non-transitory computer-readable medium of claim 1, wherein the set of

instructions that is executable by the at least one processor of the apparatus causes the apparatus to further perform:

applying a rate-distortion optimization technique to the virtual reference picture.

9. The non-transitory computer-readable medium of claim 1, wherein the set of instructions that is executable by the at least one processor of the apparatus causes the apparatus to further perform:

in response to receiving a training video sequence, determining a training prediction picture based on an optical flow between a first picture and a second picture of the training video sequence, wherein the first picture is after the second picture and before a third picture of the training video sequence in the temporal order; and

training the prediction model using the machine learning technique, wherein a plurality of pictures of the training video sequence and the training prediction picture are inputs of the prediction model, and the third picture is a label for an output of the prediction model, and wherein the plurality of pictures comprises the first picture and the second picture.

10. The non-transitory computer-readable medium of claim 9, wherein the plurality of pictures are consecutive in the temporal order.

11. The non-transitory computer-readable medium of claim 9, wherein the first picture, the second picture, and the third picture are consecutive in the temporal order.

12. The non-transitory computer-readable medium of claim 9, wherein training the prediction model comprises:

determining, in a color space, a first color component of the output of the prediction model based on first color components of the inputs of the prediction model; and

determining a second color component of the output of the prediction model based on second color components of the inputs of the prediction model.

13. The non-transitory computer-readable medium of claim 1, wherein determining the virtual reference picture comprises:

determining a synthesized background picture for the video sequence; and

determining the virtual reference picture by inputting the plurality of reconstructed pictures, the prediction picture, and the synthesized background picture into the prediction model.

14. The non-transitory computer-readable medium of claim 13, wherein the set of instructions that is executable by the at least one processor of the apparatus causes the apparatus to further perform:

in response to receiving a training video sequence, determining a training prediction picture based on an optical flow between a first picture and a second picture of the training video sequence, wherein the first picture is after the second picture and before a third picture of the training video sequence in the temporal order;

receiving a background picture of the training video sequence, wherein the background picture comprises no foreground object; and

training the prediction model using the machine learning technique, wherein a plurality of pictures of the training video sequence, the training prediction picture, and the background picture are inputs of the prediction model, and the third picture is a label for an output of the prediction model, and wherein the plurality of pictures comprises the first

picture and the second picture.

15. An apparatus, comprising:

a memory configured to store a set of instructions; and

one or more processors communicatively coupled to the memory and configured to execute the set of instructions to cause the apparatus to perform:

in response to receiving a video sequence, determining a plurality of reconstructed pictures before a target picture of the video sequence in a temporal order;

determining a prediction picture based on an optical flow between two of the plurality of reconstructed pictures;

determining a virtual reference picture by inputting the plurality of reconstructed pictures and the prediction picture into a prediction model determined using a machine learning technique; and

encoding or decoding the target picture using the virtual reference picture as a reference picture.

16. The apparatus of claim 15, wherein determining the prediction picture based on the optical flow between the two of the plurality of reconstructed pictures comprises:

determining a first optical flow from a first reconstructed picture to a second reconstructed picture, wherein the two of the plurality of reconstructed pictures comprises the first reconstructed picture and the second reconstructed picture, and the first reconstructed picture is after the second reconstructed picture and before the target picture in the temporal order;

determining a second optical flow from the first reconstructed picture to the target

picture based on the first optical flow;

determining a third optical flow from the target picture to the first reconstructed picture based on the second optical flow; and

determining, using a backward warping technique, the prediction picture based on the first reconstructed picture and the third optical flow.

17. The apparatus of claim 15, wherein the one or more processors are further configured to execute the set of instructions to cause the apparatus to perform:

in response to receiving a training video sequence, determining a training prediction picture based on an optical flow between a first picture and a second picture of the training video sequence, wherein the first picture is after the second picture and before a third picture of the training video sequence in the temporal order; and

training the prediction model using the machine learning technique, wherein a plurality of pictures of the training video sequence and the training prediction picture are inputs of the prediction model, and the third picture is a label for an output of the prediction model, and wherein the plurality of pictures comprises the first picture and the second picture.

18. The apparatus of claim 15, wherein determining the virtual reference picture comprises:

determining a synthesized background picture for the video sequence; and

determining the virtual reference picture by inputting the plurality of reconstructed pictures, the prediction picture, and the synthesized background picture into the prediction model.

19. The apparatus of claim 18, wherein the one or more processors are further configured to execute the set of instructions to cause the apparatus to perform:

in response to receiving a training video sequence, determining a training prediction picture based on an optical flow between a first picture and a second picture of the training video sequence, wherein the first picture is after the second picture and before a third picture of the training video sequence in the temporal order;

receiving a background picture of the training video sequence, wherein the background picture comprises no foreground object; and

training the prediction model using the machine learning technique, wherein a plurality of pictures of the training video sequence, the training prediction picture, and the background picture are inputs of the prediction model, and the third picture is a label for an output of the prediction model, and wherein the plurality of pictures comprises the first picture and the second picture.

20. A method, comprising:

in response to receiving a video sequence, determining a plurality of reconstructed pictures before a target picture of the video sequence in a temporal order;

determining a prediction picture based on an optical flow between two of the plurality of reconstructed pictures;

determining a virtual reference picture by inputting the plurality of reconstructed pictures and the prediction picture into a prediction model determined using a machine learning technique; and

encoding or decoding the target picture using the virtual reference picture as a reference picture.

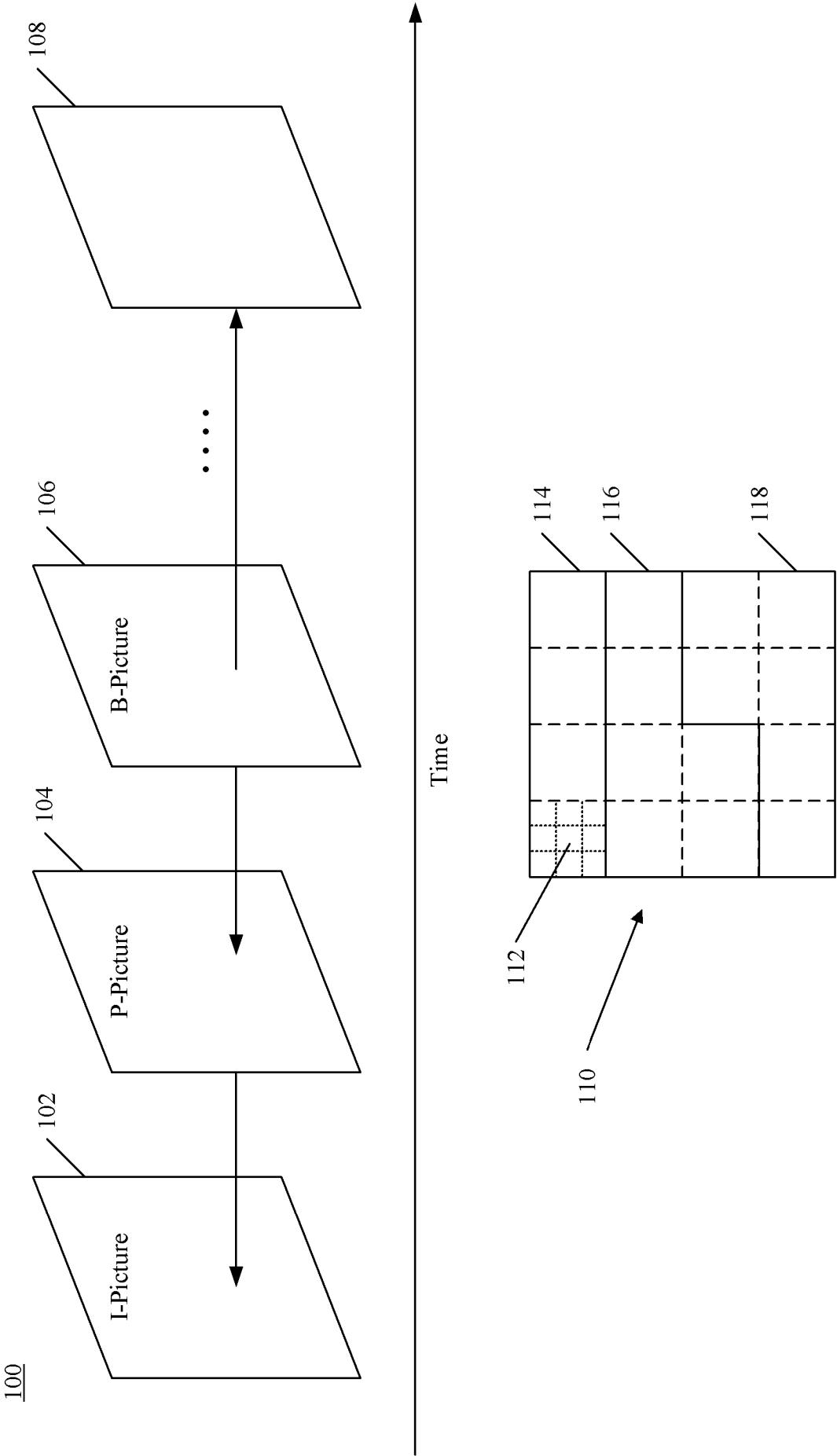


FIG. 1

200A

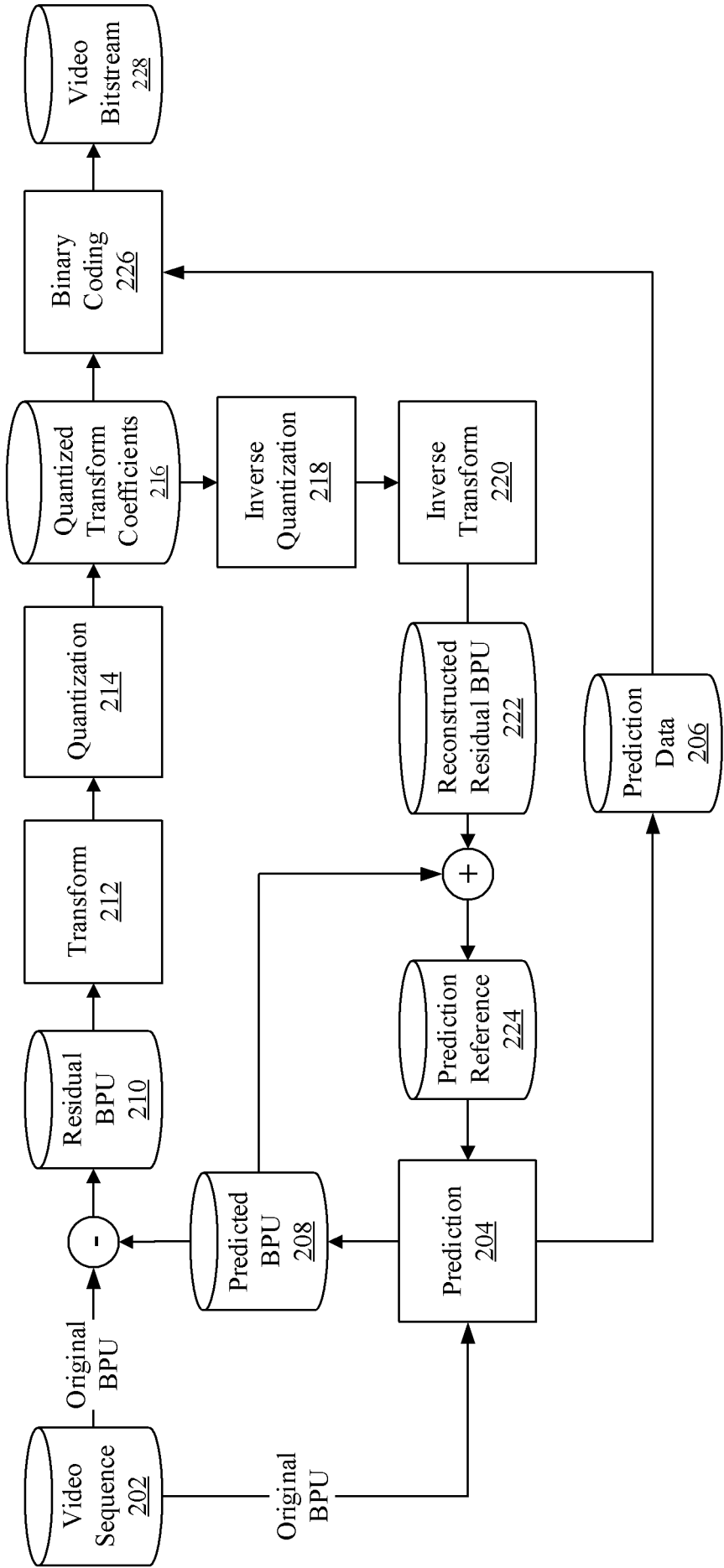


FIG. 2A

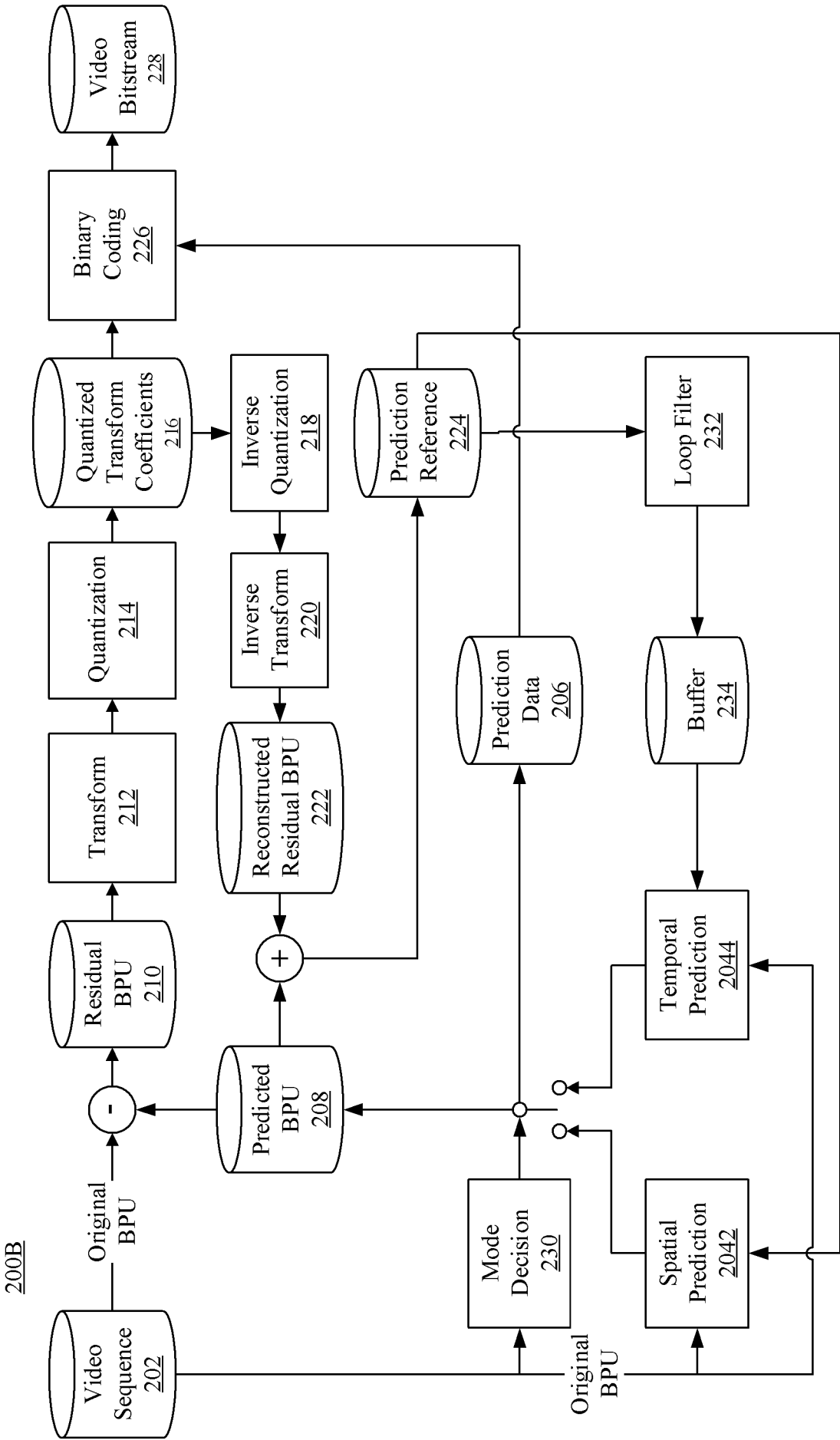


FIG. 2B

300A

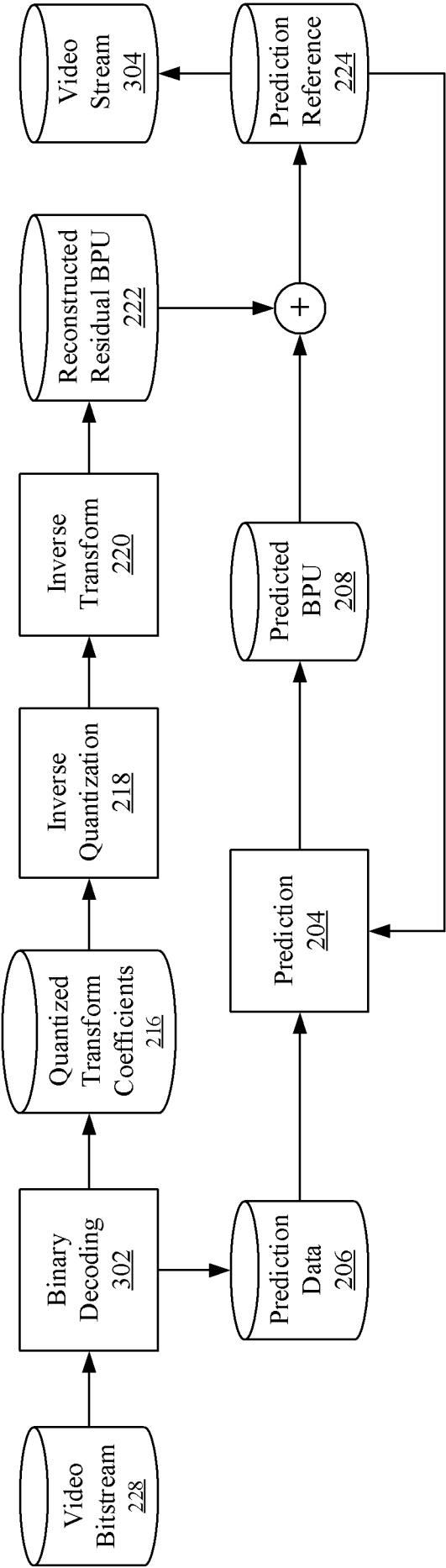


FIG. 3A

300B

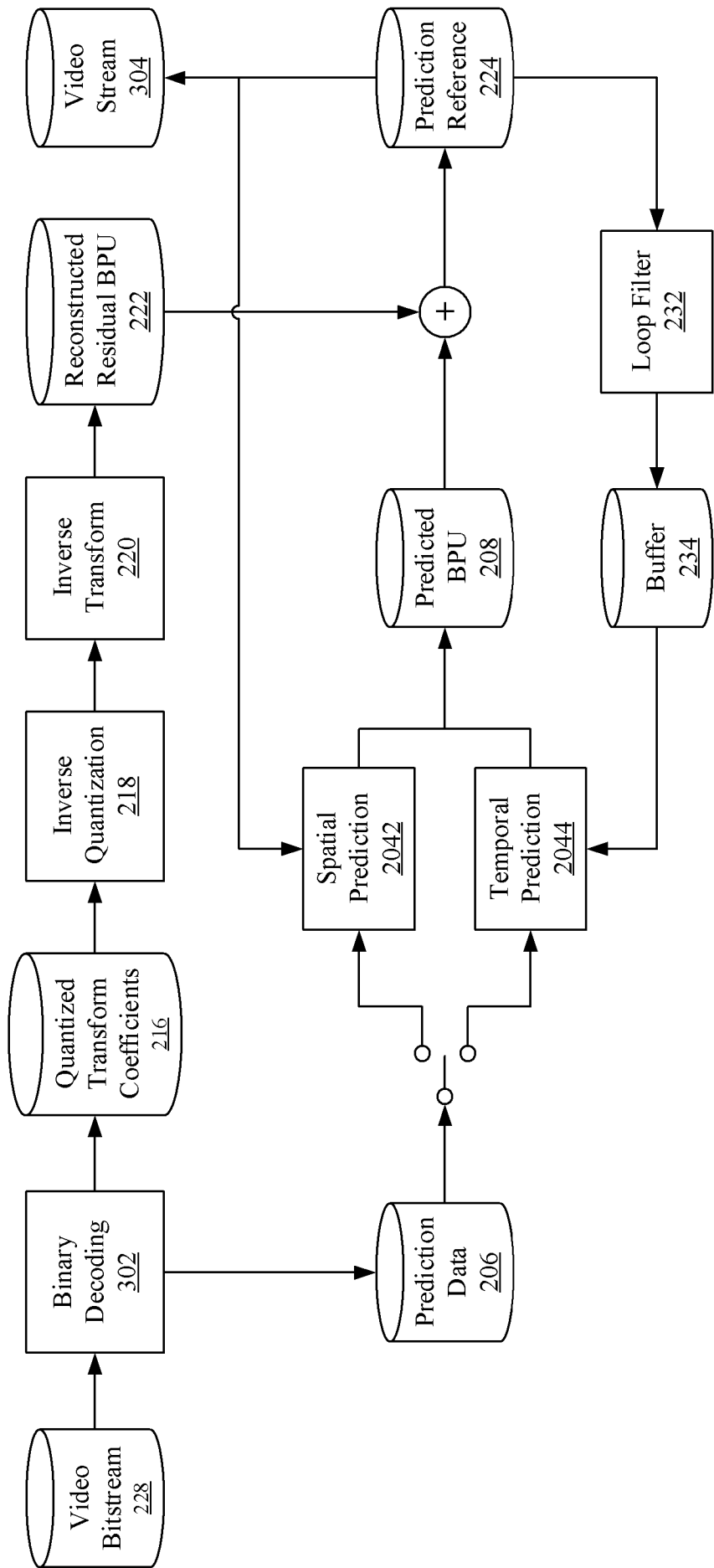


FIG. 3B

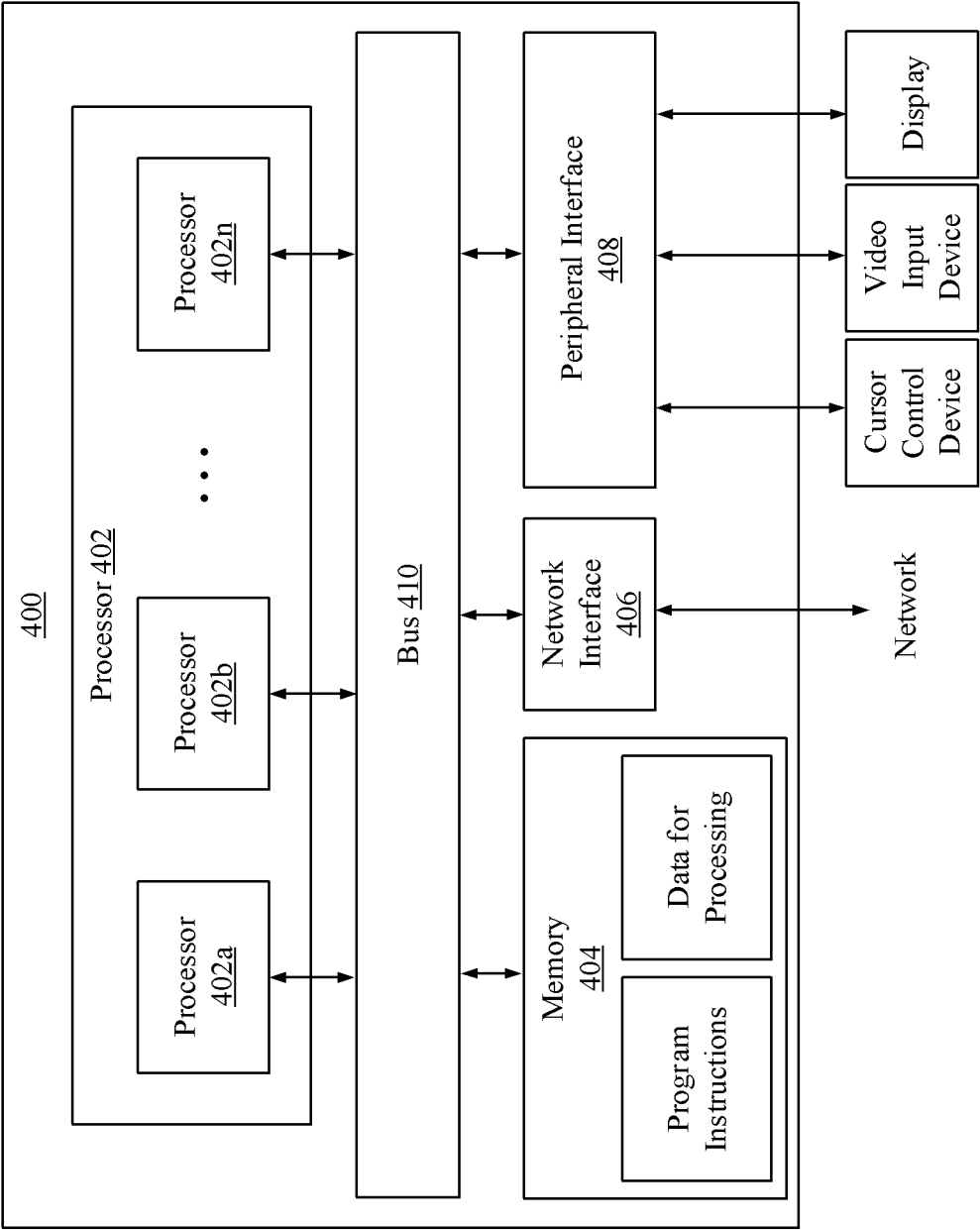


FIG. 4

500

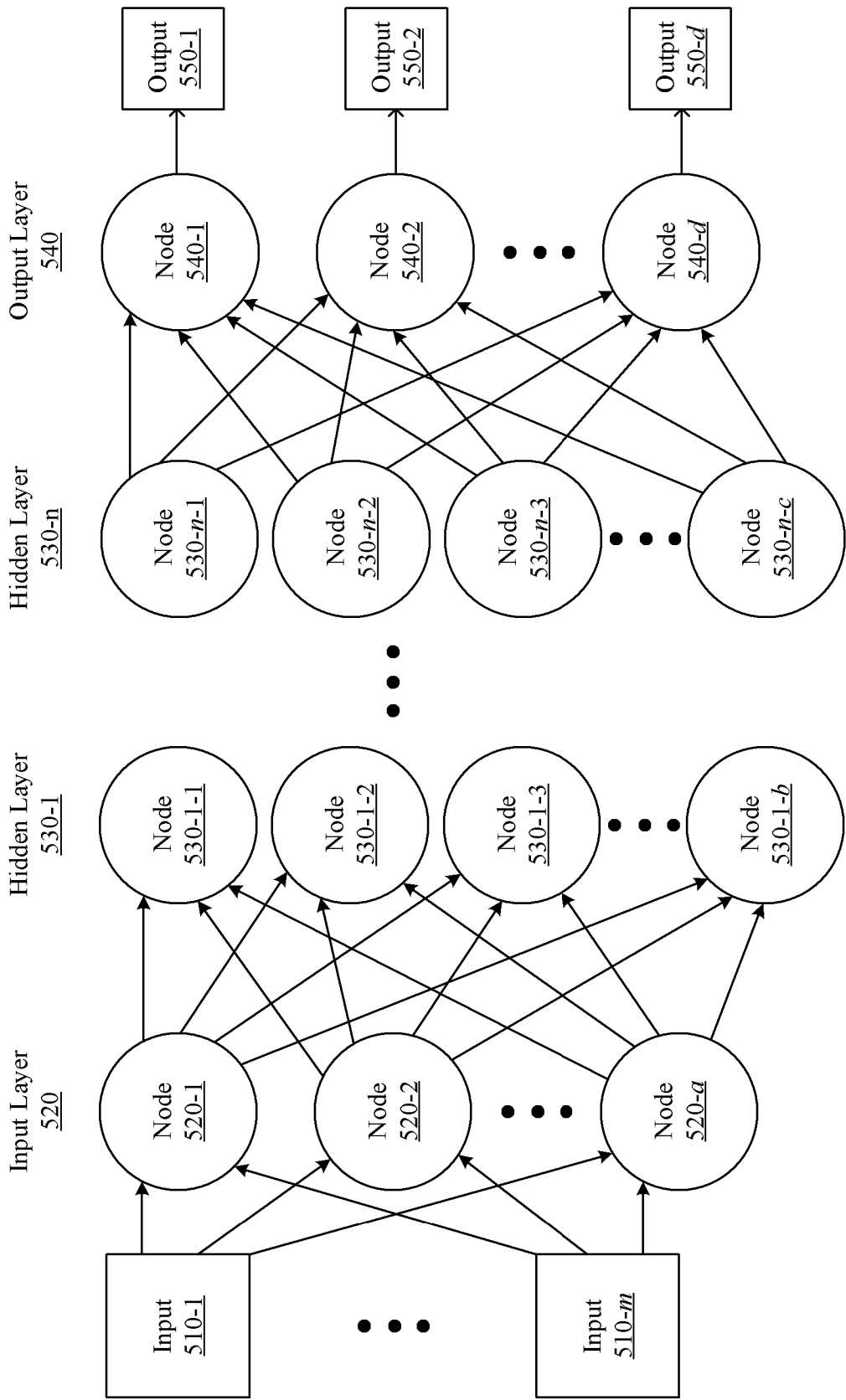


FIG. 5

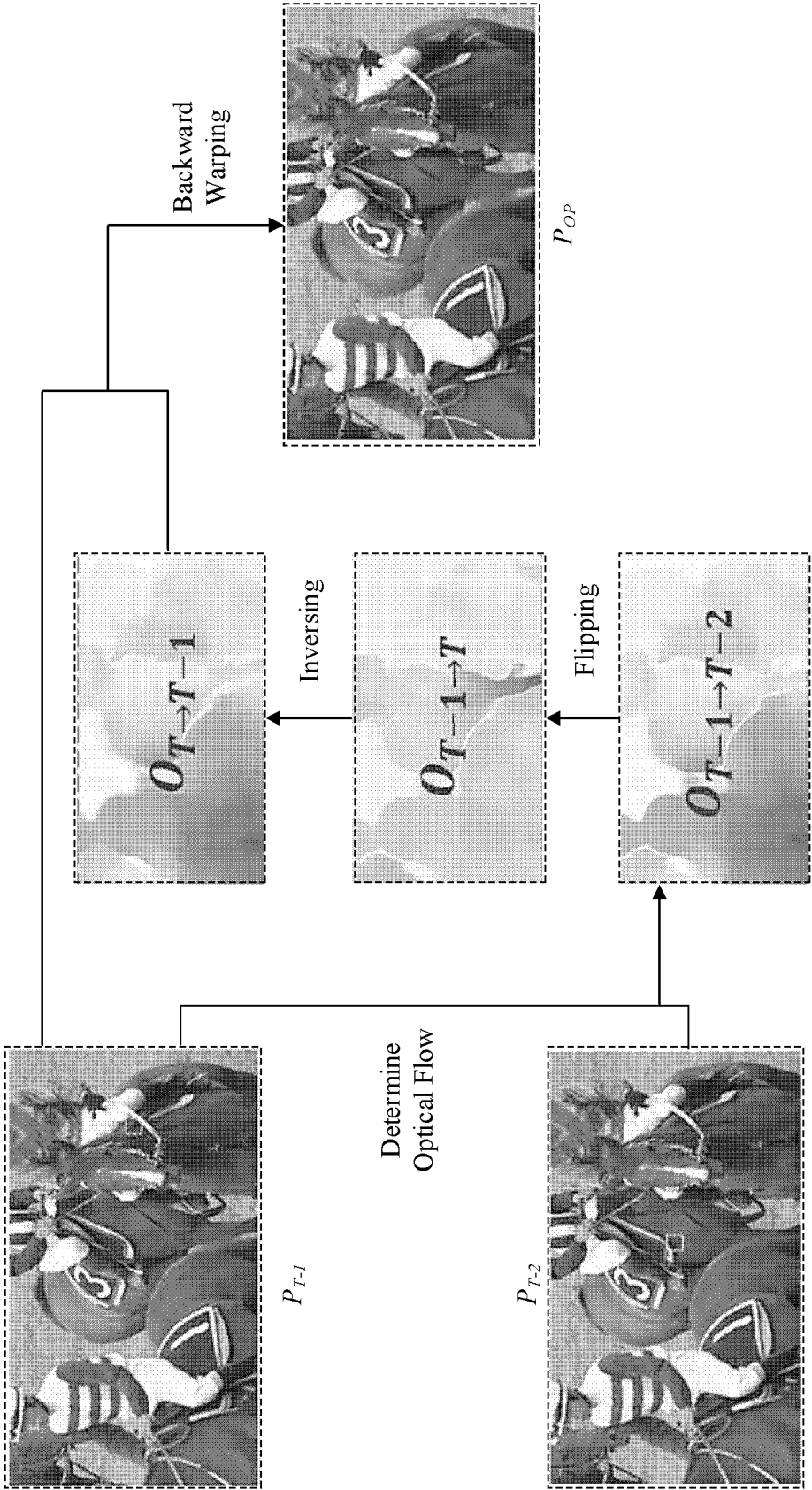


FIG. 6

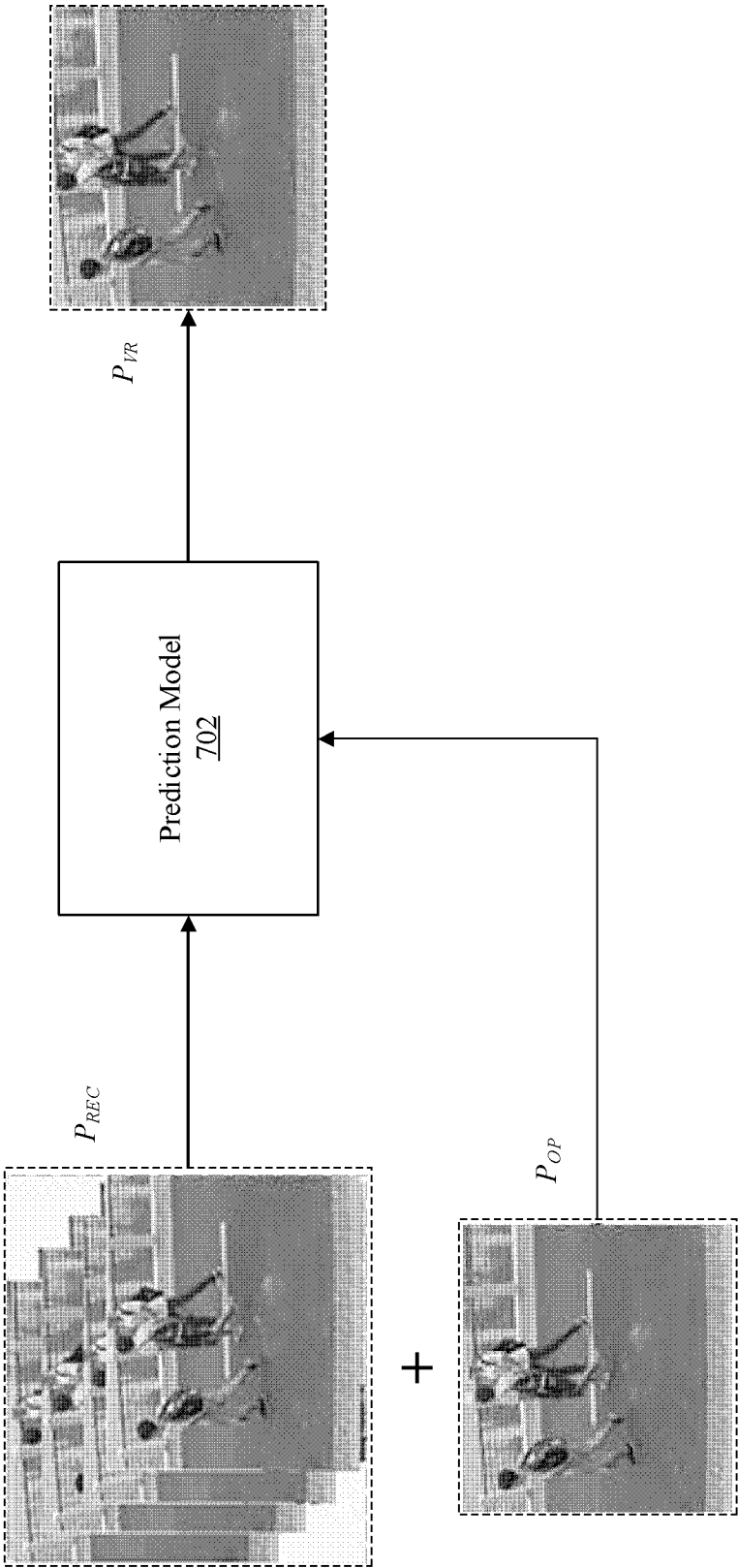


FIG. 7

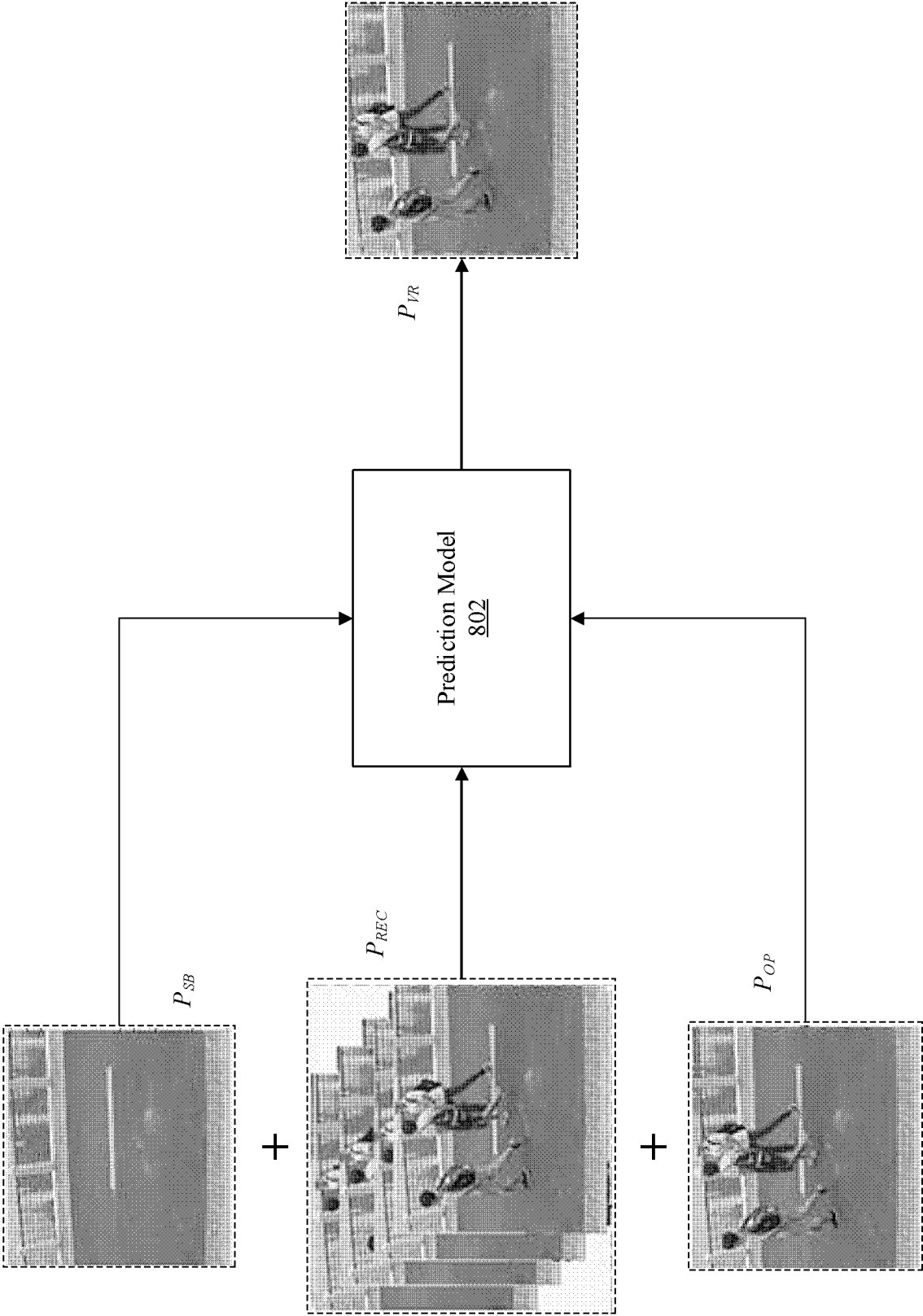


FIG. 8

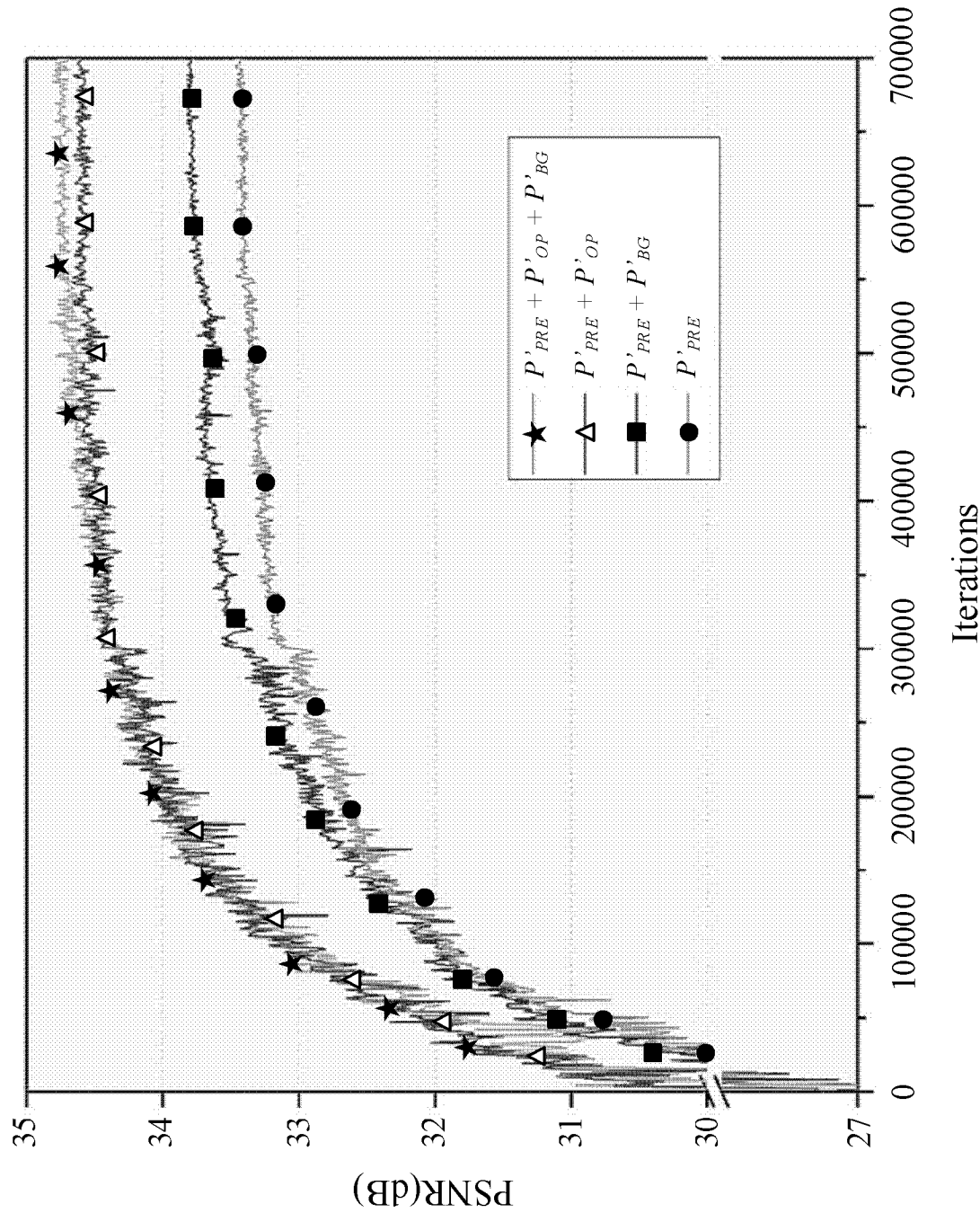
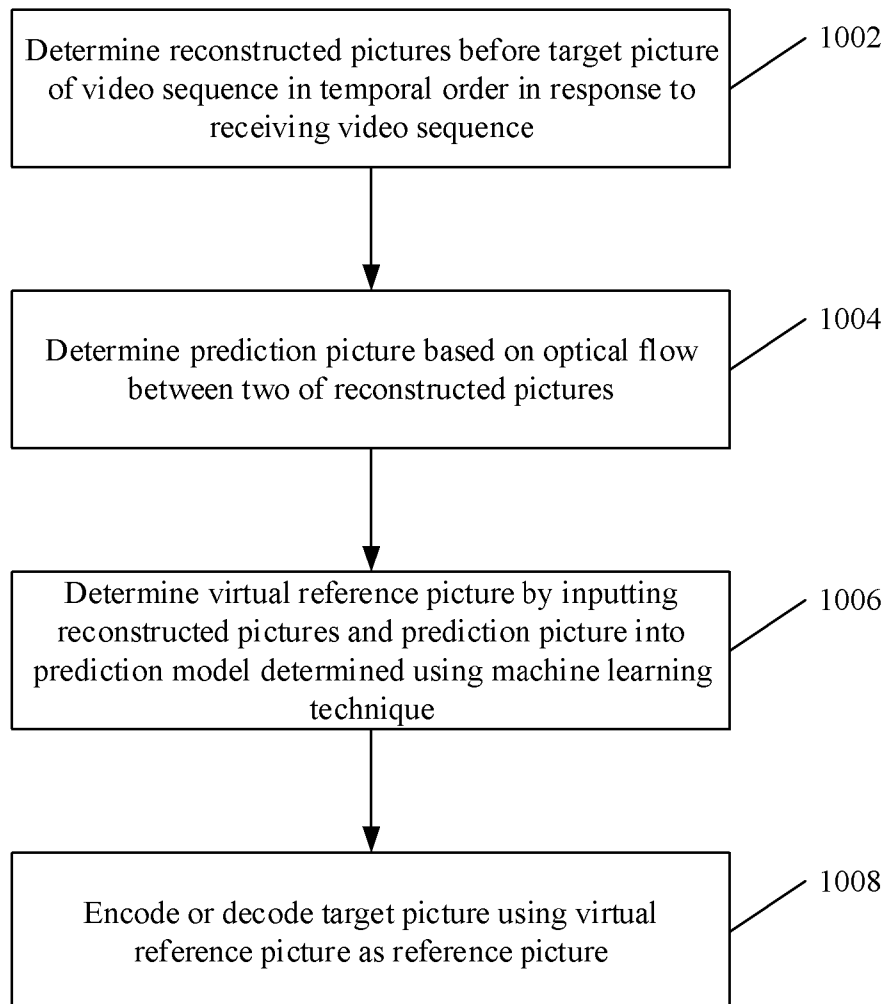


FIG. 9

1000**FIG. 10**

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2020/129887

A. CLASSIFICATION OF SUBJECT MATTER

H04N 19/513(2014.01)i

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04N

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

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

CNPAT, CNKI, WPI, EPODOC, IEEE: video, encode, decode, code, reference, picture, frame, image, reconstruct, predict, target, optical flow, virtual, model, machine learning, temporal order

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2019306526 A1 (ELECTRONICS AND TELECOMMUNICATIONS RESEARCH INSTITUTE) 03 October 2019 (2019-10-03) description, paragraphs 627-662 and 753-822	1-20
A	US 10841577 B2 (ELECTRONICS AND TELECOMMUNICATIONS RESEARCH INSTITUTE) 17 November 2020 (2020-11-17) the whole document	1-20
A	US 2016165237 A1 (QUALCOMM INCORPORATED) 09 June 2016 (2016-06-09) the whole document	1-20
A	DING, Qing et al. "VRFCNN: Virtual Reference Frame Generation Network for Quality SHVC" <i>IEEE SIGNAL PROCESSING LETTERS</i> , Vol. 27, 16 November 2020 (2020-11-16), the whole document	1-20
A	US 2020359041 A1 (CISCO TECHNOLOGY, INC.) 12 November 2020 (2020-11-12) the whole 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

"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

19 July 2021

Date of mailing of the international search report

17 August 2021

Name and mailing address of the ISA/CN

National Intellectual Property Administration, PRC
6, Xitucheng Rd., Jimen Bridge, Haidian District, Beijing
100088
China

Facsimile No. (86-10)62019451

Authorized officer

HU,Di

Telephone No. 86-(10)-53961704

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2020/129887

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
US	2019306526	A1	03 October 2019	None			
US	10841577	B2	17 November 2020	US	2019246102	A1	08 August 2019
US	2016165237	A1	09 June 2016	ZA	201403984	B	28 June 2017
				CA	2852959	A1	10 May 2013
				JP	5837216	B2	24 December 2015
				KR	101597927	B1	25 February 2016
				AU	2016201877	A1	21 April 2016
				EP	2774365	B1	22 August 2018
				CN	103947210	B	31 October 2017
				IN	3181CHN2014	A	03 July 2015
				RU	2014122182	A	10 December 2015
				CN	103947210	A	23 July 2014
				CA	2852959	C	05 December 2017
				KR	20140088592	A	10 July 2014
				US	2013107953	A1	02 May 2013
				MY	167629	A	21 September 2018
				WO	2013067033	A1	10 May 2013
				AU	2012332567	A1	22 May 2014
				US	9264717	B2	16 February 2016
				SG	11201401440X	A	27 June 2014
				EP	2774365	A1	10 September 2014
				BR	112014010418	A2	25 April 2017
				JP	2014535219	A	25 December 2014
US	2020359041	A1	12 November 2020	US	2017111652	A1	20 April 2017
				WO	2017066235	A1	20 April 2017
				US	10805627	B2	13 October 2020