



US 20250267282A1

(19) **United States**

(12) **Patent Application Publication**
LI et al.

(10) **Pub. No.: US 2025/0267282 A1**

(43) **Pub. Date: Aug. 21, 2025**

(54) **METHOD, APPARATUS, AND MEDIUM FOR VIDEO PROCESSING**

(71) Applicants: **Douyin Vision Co., Ltd.**, Beijing (CN);
Bytedance Inc., Los Angeles, CA (US)

(72) Inventors: **Junru LI**, Beijing (CN); **Kai ZHANG**,
Los Angeles, CA (US); **Li ZHANG**,
Los Angeles, CA (US)

(21) Appl. No.: **19/177,459**

(22) Filed: **Apr. 11, 2025**

Related U.S. Application Data

(63) Continuation of application No. PCT/CN2023/
124366, filed on Oct. 12, 2023.

(30) **Foreign Application Priority Data**

Oct. 13, 2022 (WO) PCT/CN2022/125228

Publication Classification

(51) **Int. Cl.**

H04N 19/147 (2014.01)

H04N 19/117 (2014.01)

H04N 19/184 (2014.01)

(52) **U.S. Cl.**

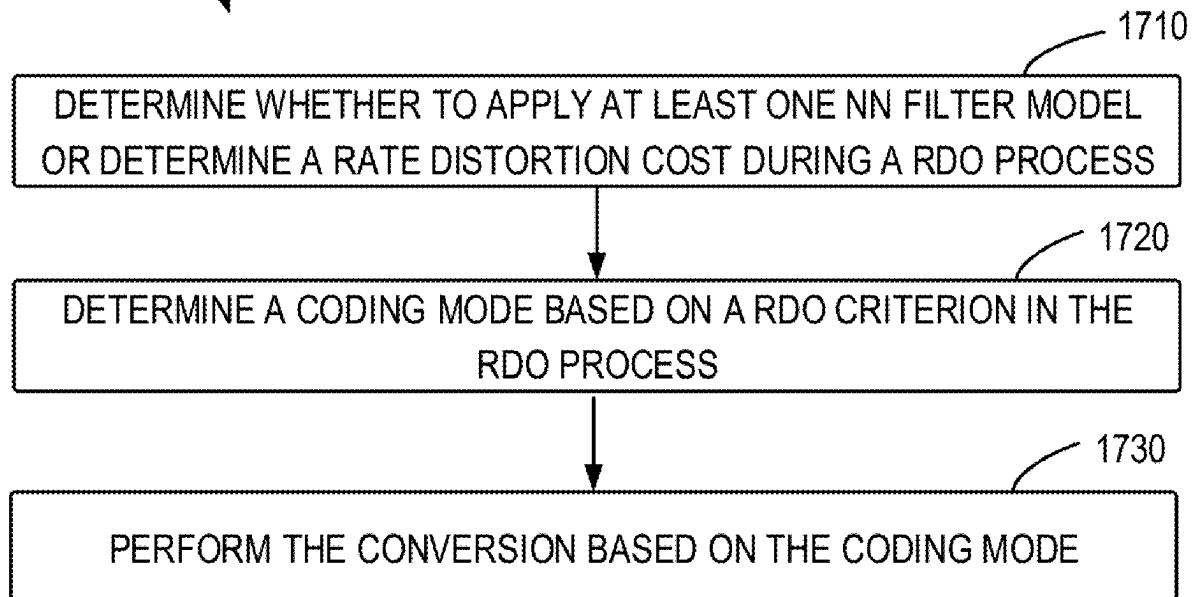
CPC **H04N 19/147** (2014.11); **H04N 19/117**
(2014.11); **H04N 19/184** (2014.11)

(57)

ABSTRACT

Embodiments of the present disclosure provide a solution for video processing. A method for video processing is proposed. The method comprises: determining, for a conversion between a video unit of a video and a bitstream of the video unit, whether to apply at least one neural network (NN) filter model or determine a rate distortion cost during a rate distortion optimization (RDO) process of the video unit based on at least one of: a distortion without NN filter model, a distortion with n-th NN filter model, a combination of distortions of a plurality of NN filter models, or coding statistics of the video unit, and wherein n is an integer number; determining a coding mode of the video unit based on a rate distortion optimization (RDO) criterion in the RDO process; and performing the conversion based on the coding mode.

1700



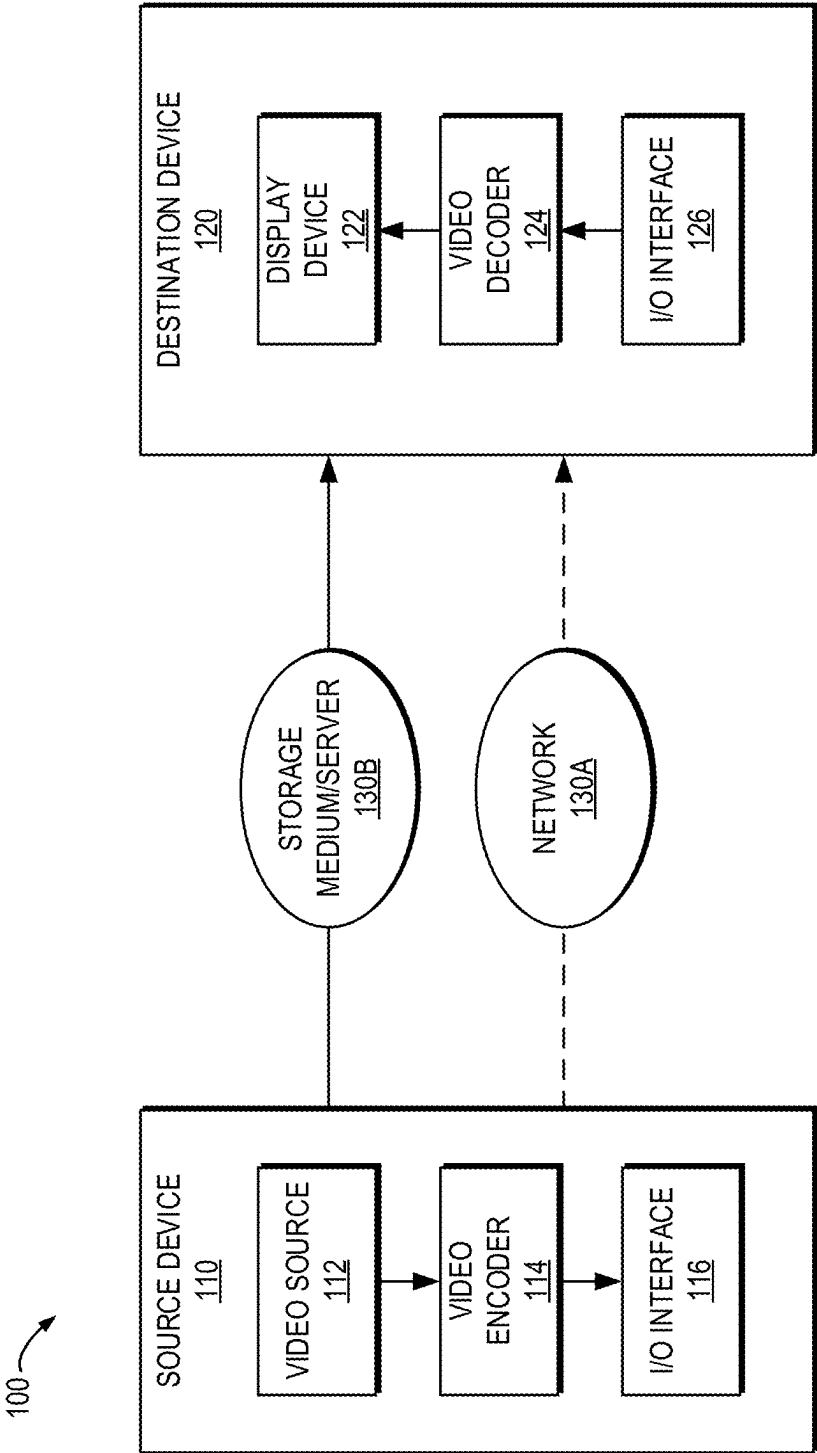


Fig. 1

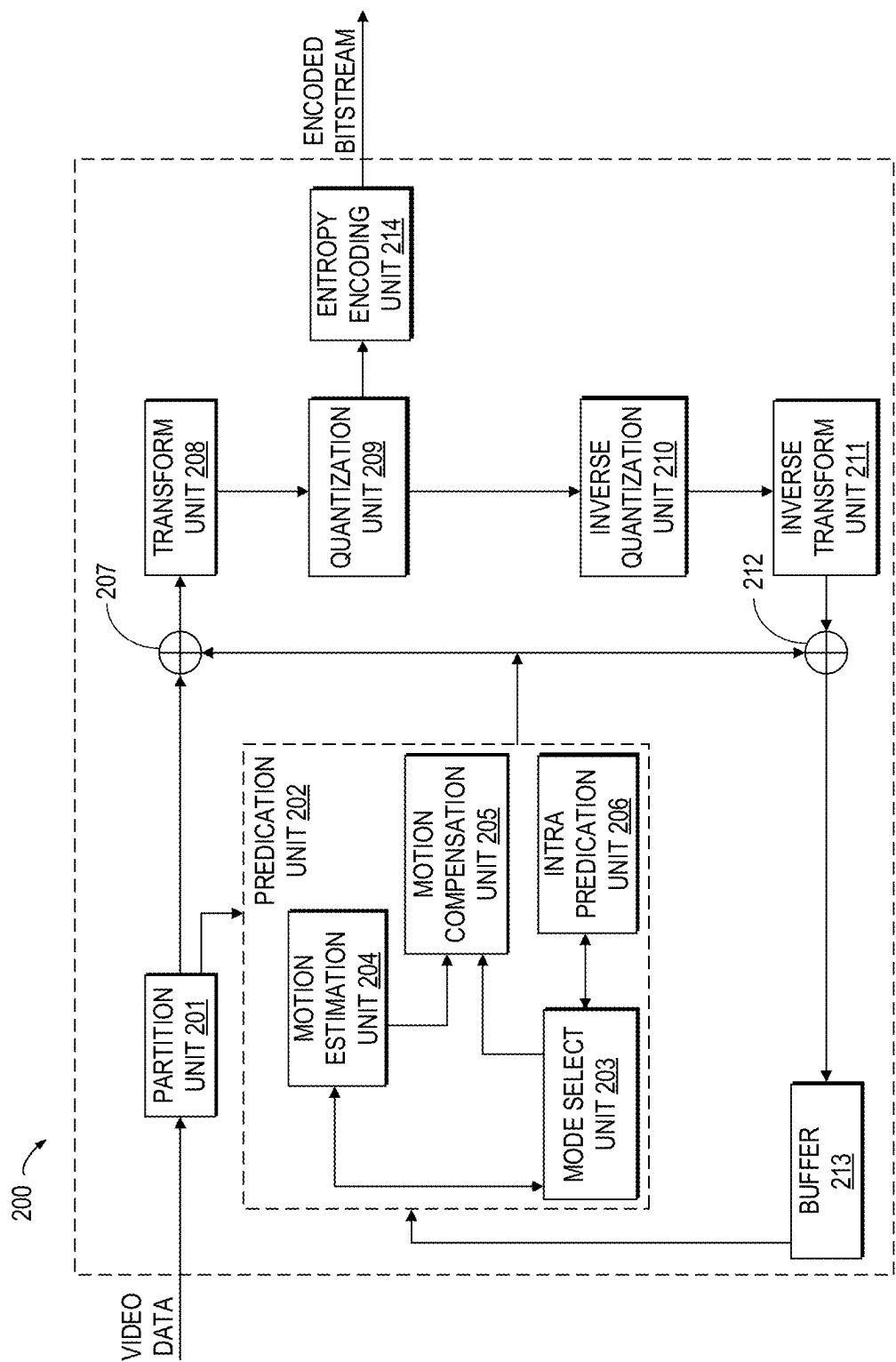


Fig. 2

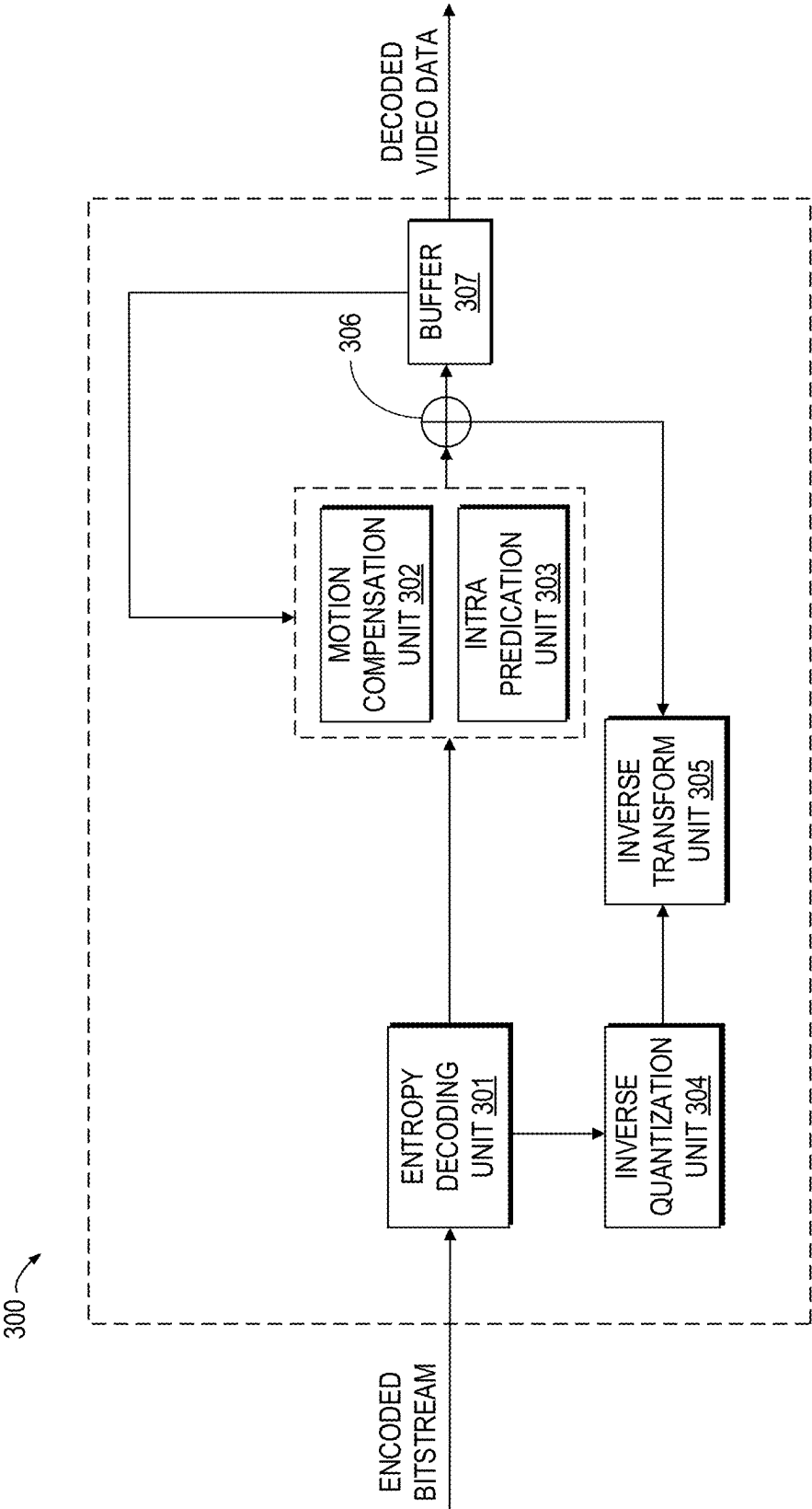


Fig. 3

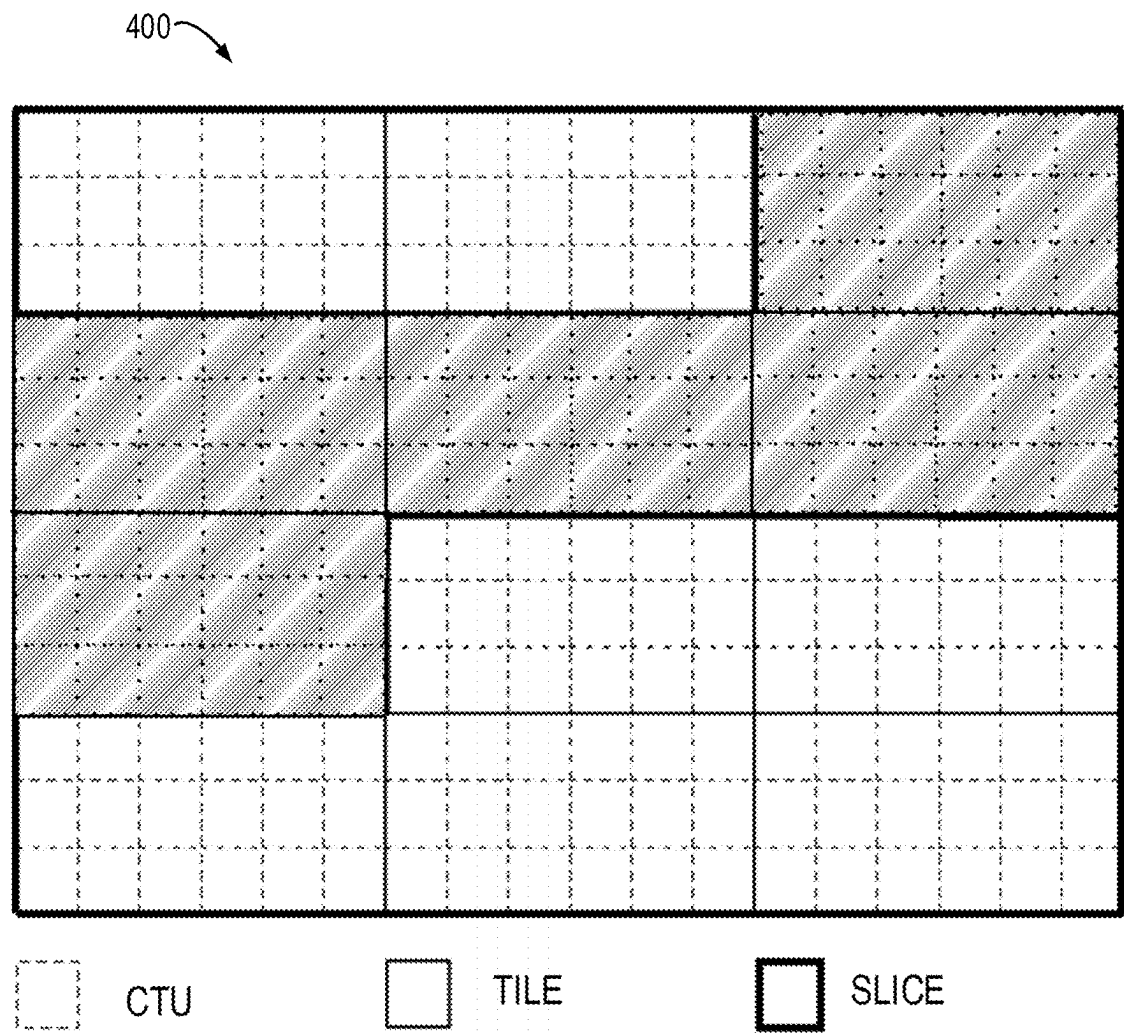


Fig. 4

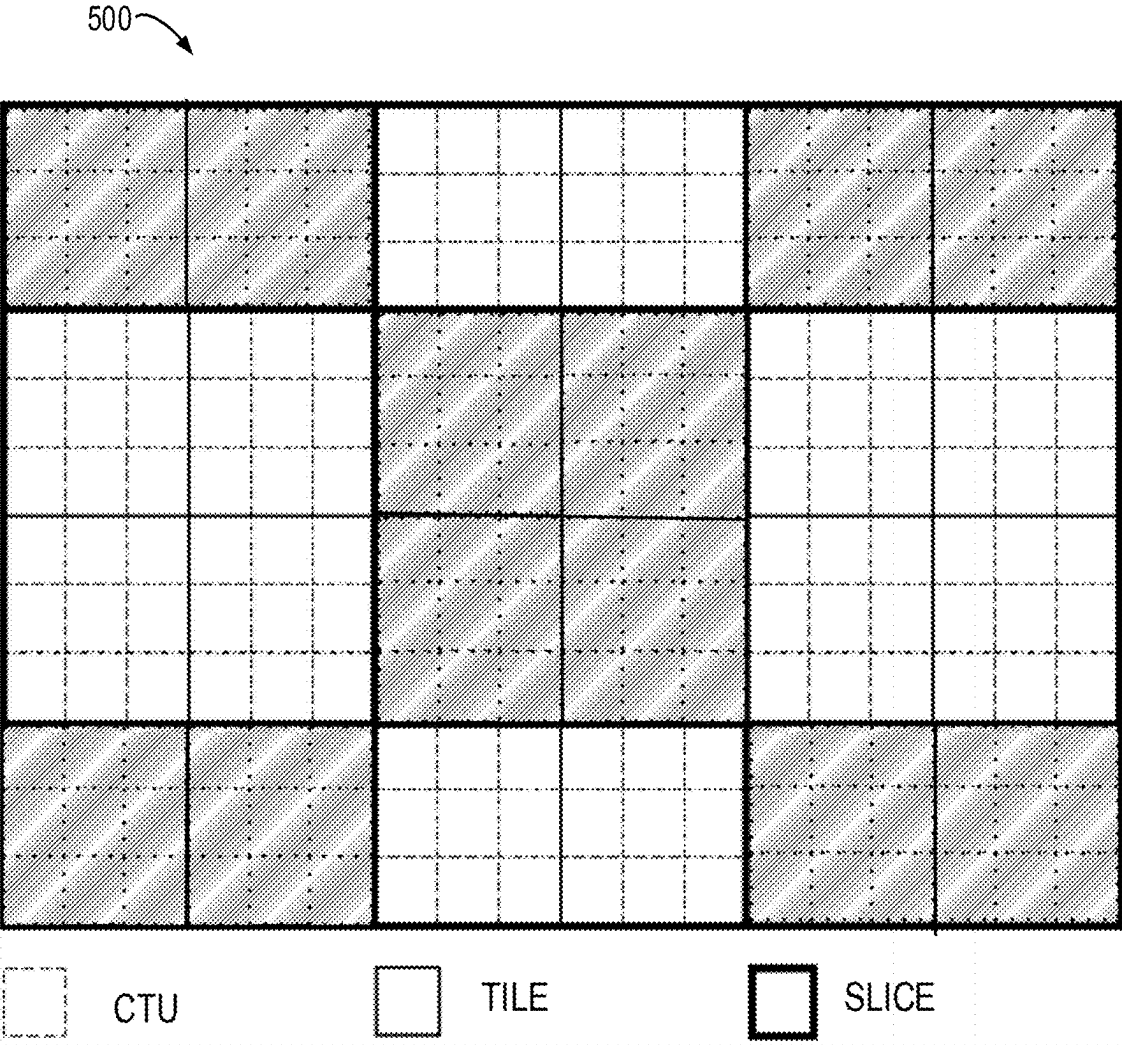


Fig. 5

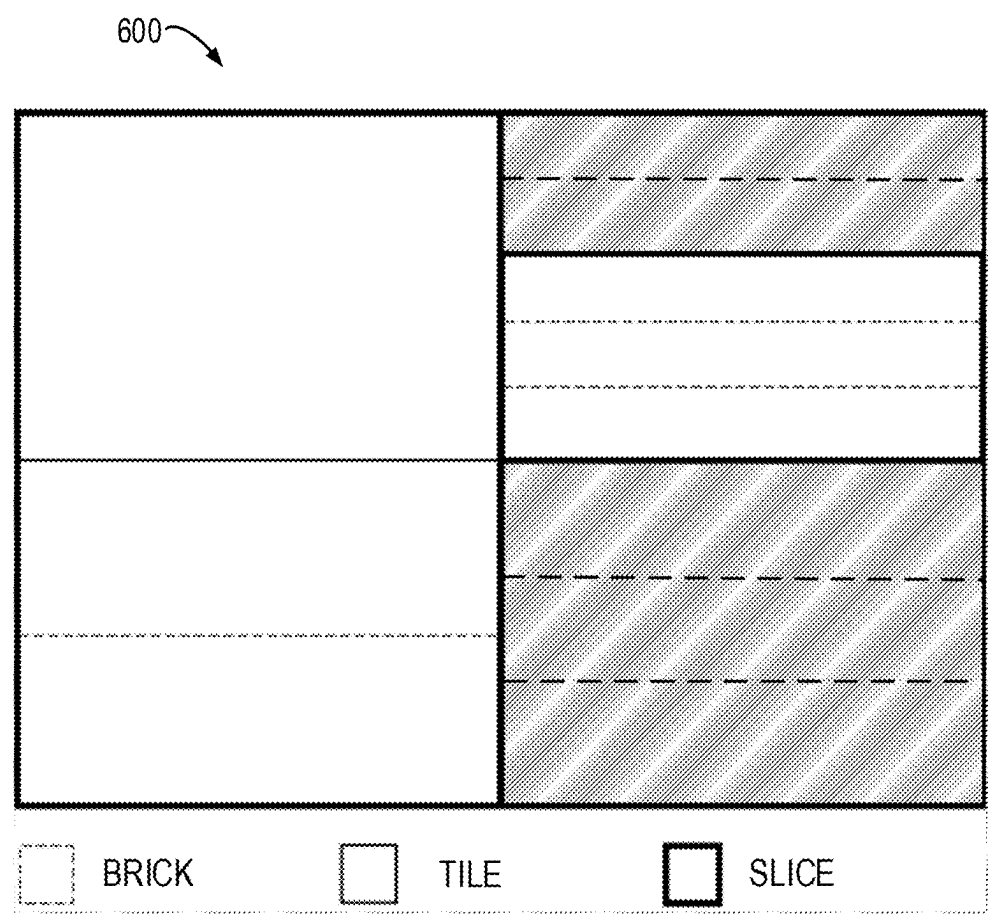


Fig. 6

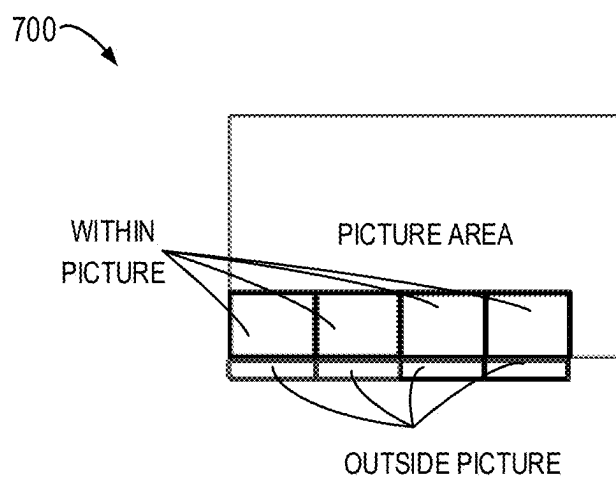


Fig. 7A

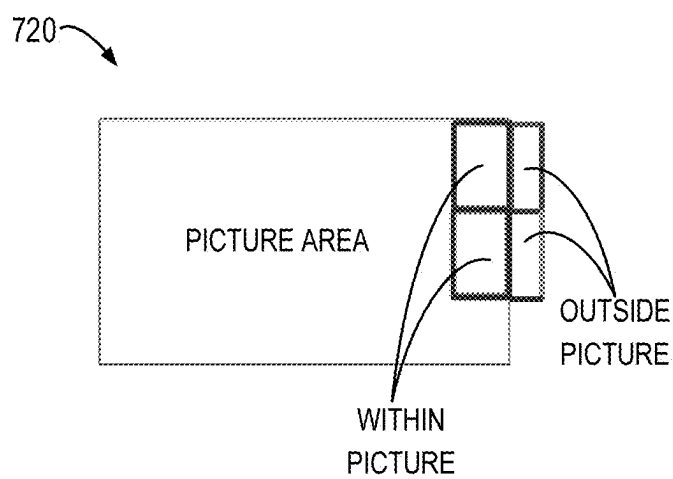


Fig. 7B

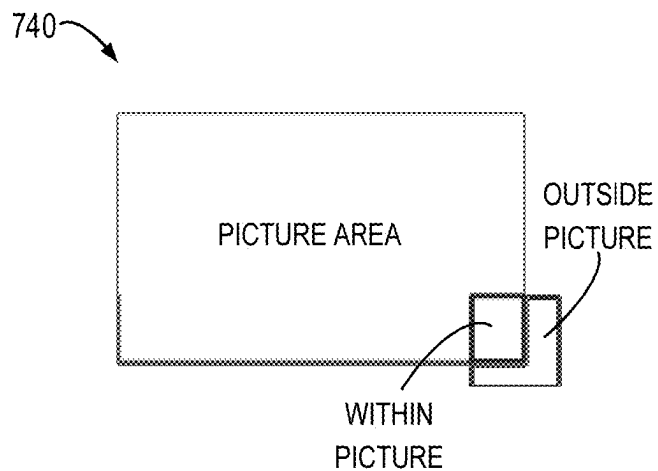


Fig. 7C

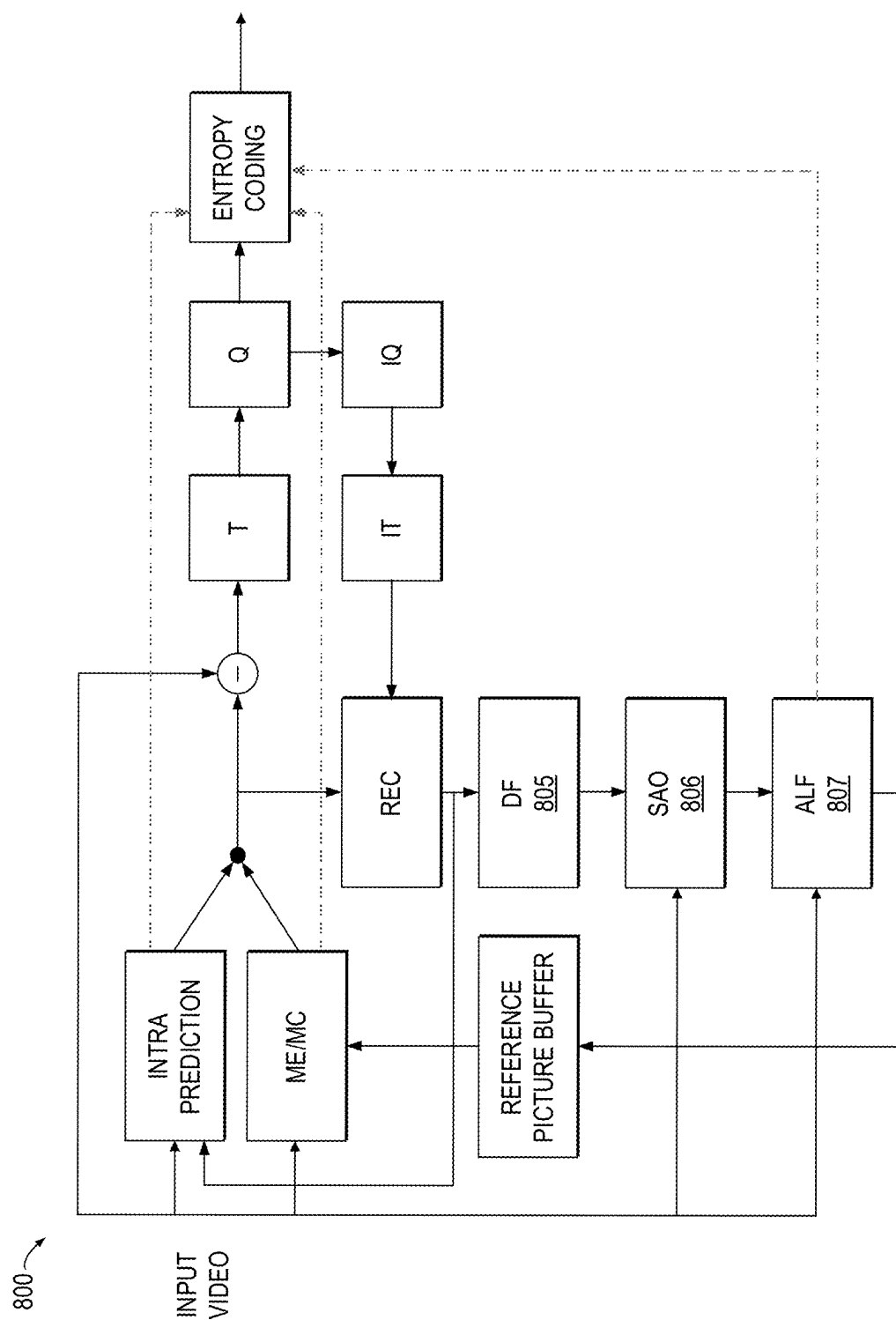


Fig. 8

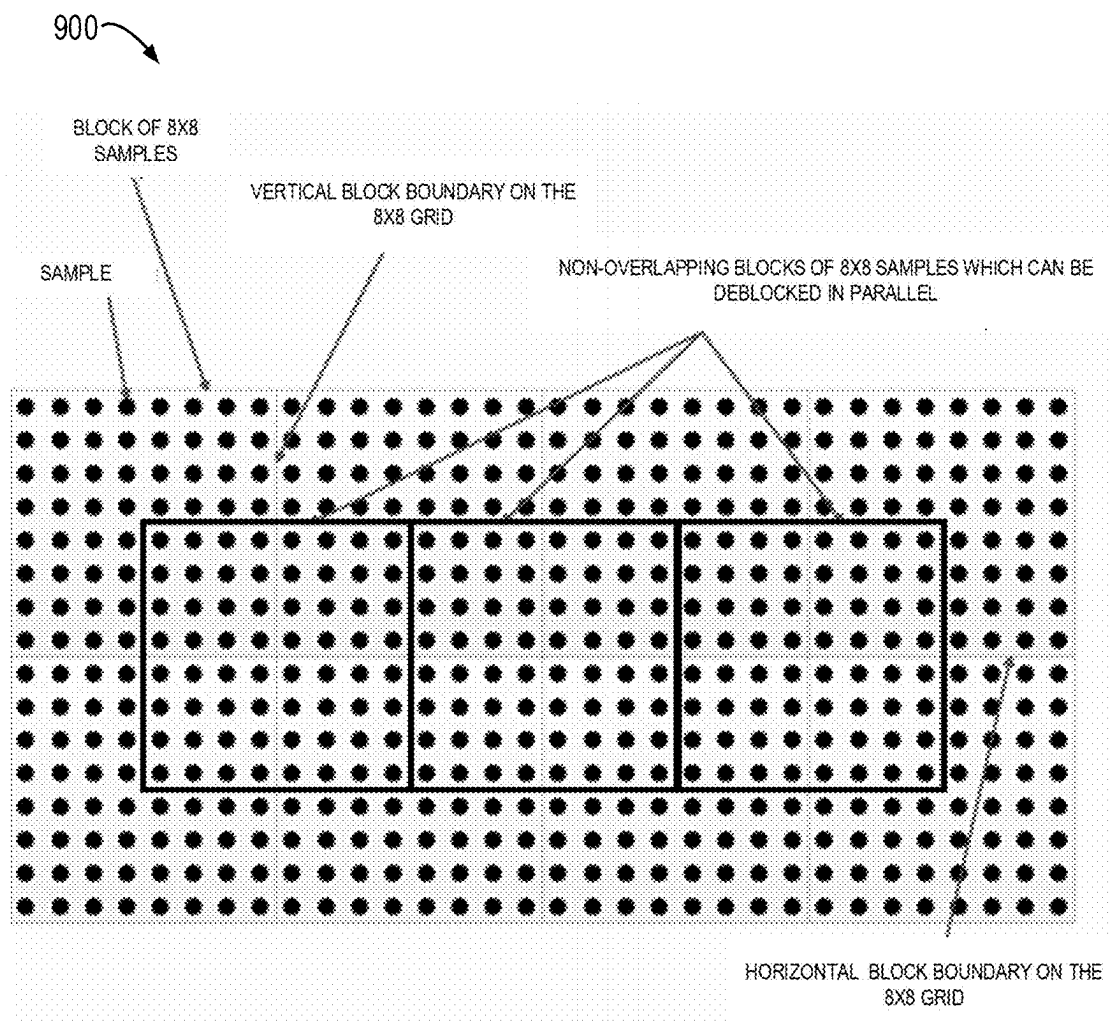


Fig. 9

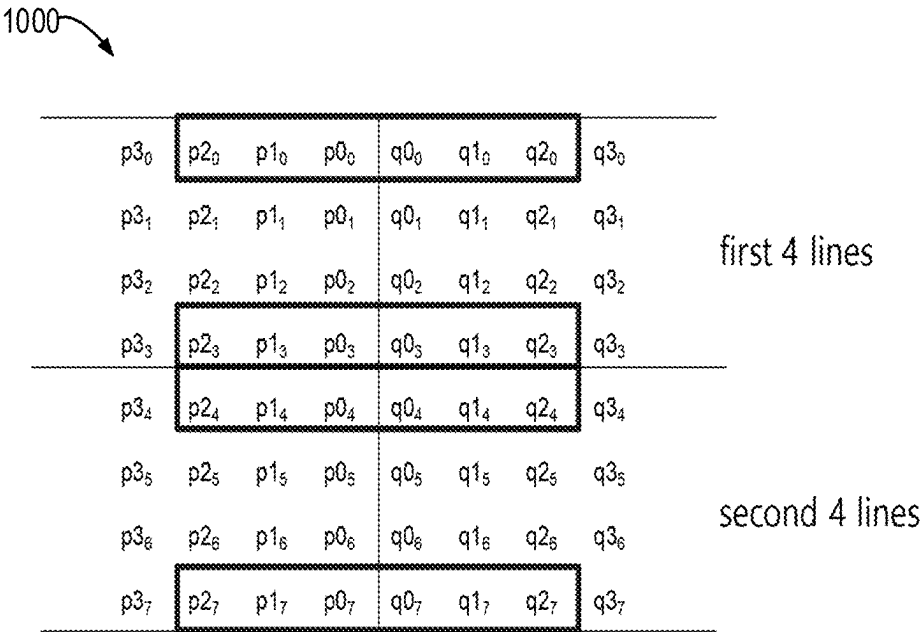
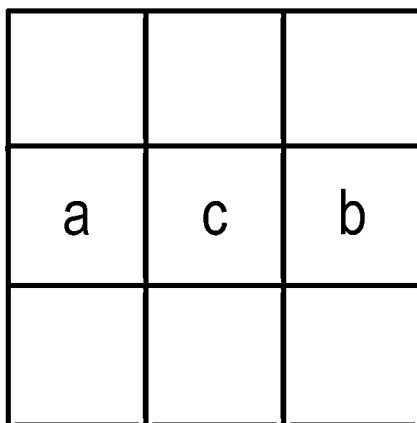


Fig. 10

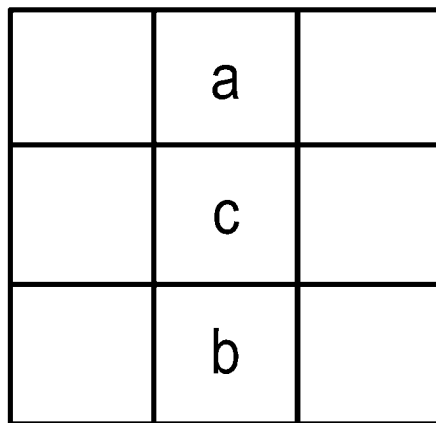
1100



a	c	b

Fig. 11A

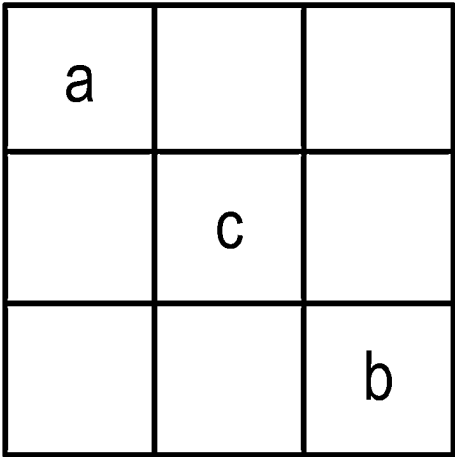
1120



	a	
	c	
	b	

Fig. 11B

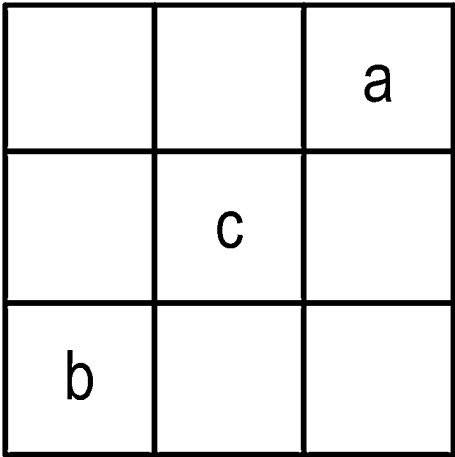
1140



a		
	c	
		b

Fig. 11C

1160



		a
	c	
b		

Fig. 11D

1200

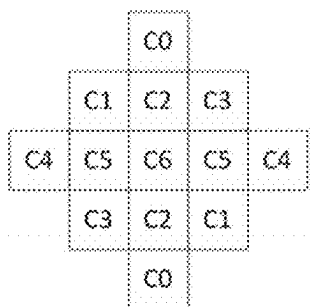


Fig. 12A

1220

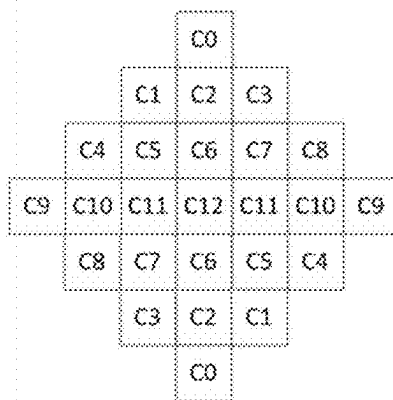


Fig. 12B

1240

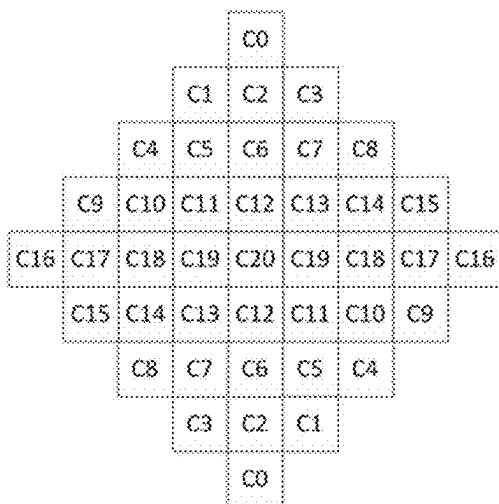


Fig. 12C

1300

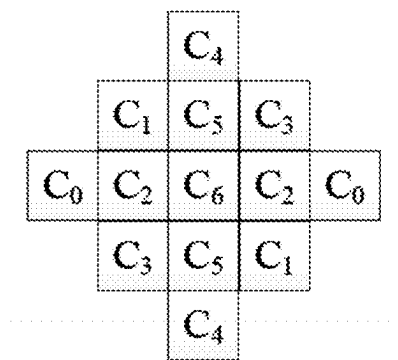


Fig. 13A

1320

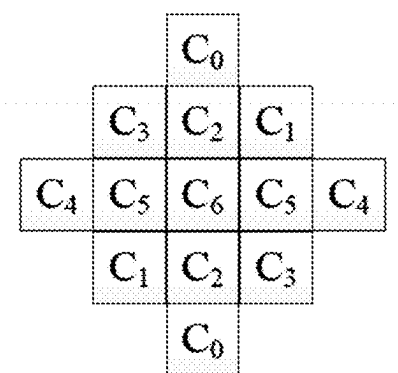


Fig. 13B

1340

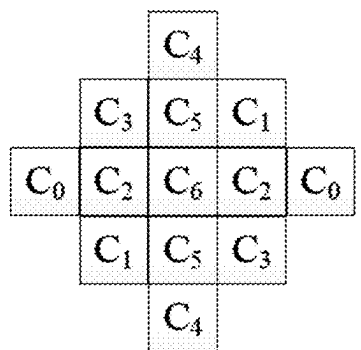


Fig. 13C

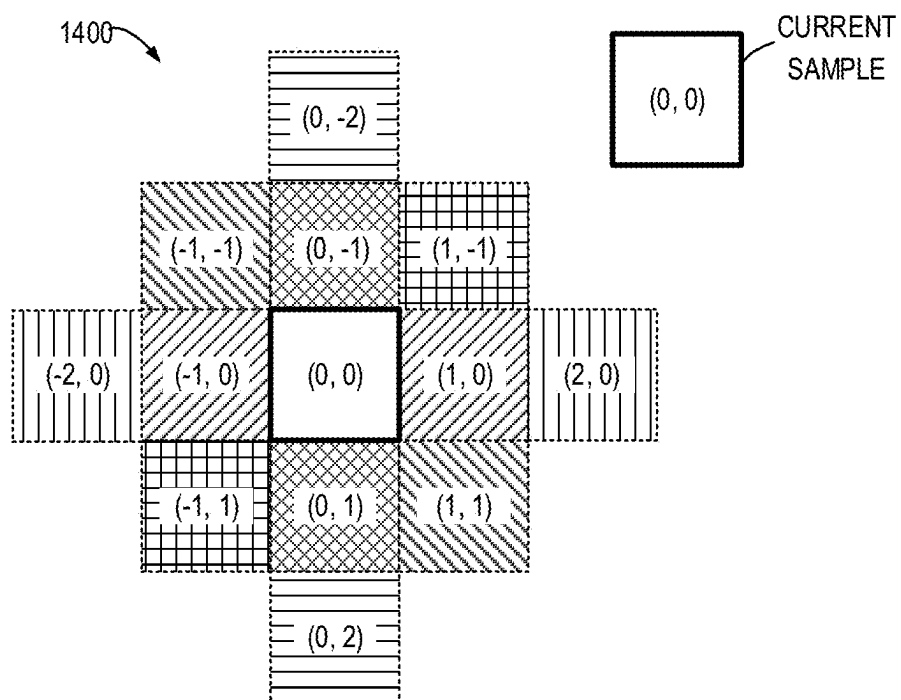


Fig. 14

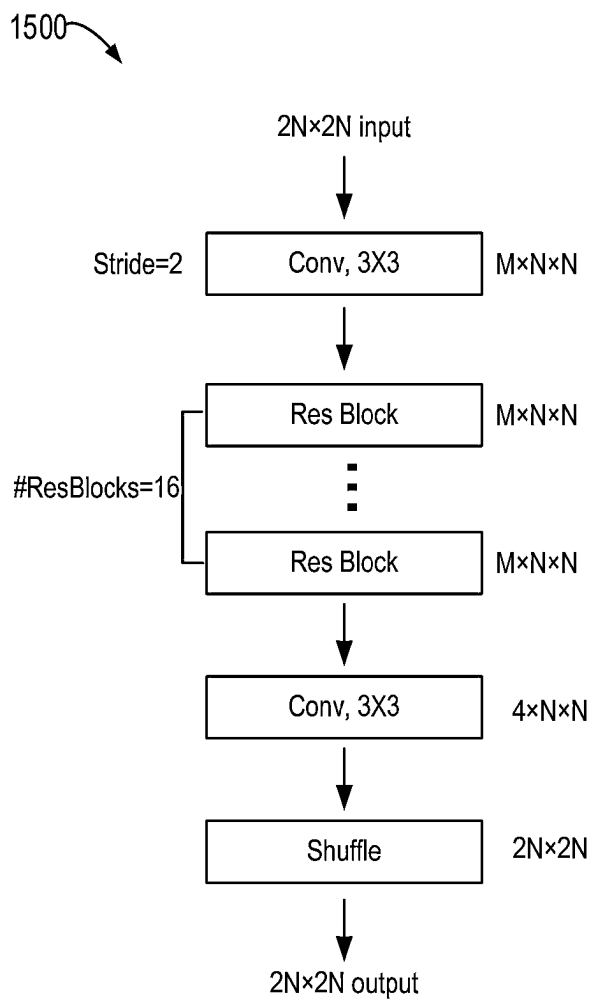


Fig. 15A

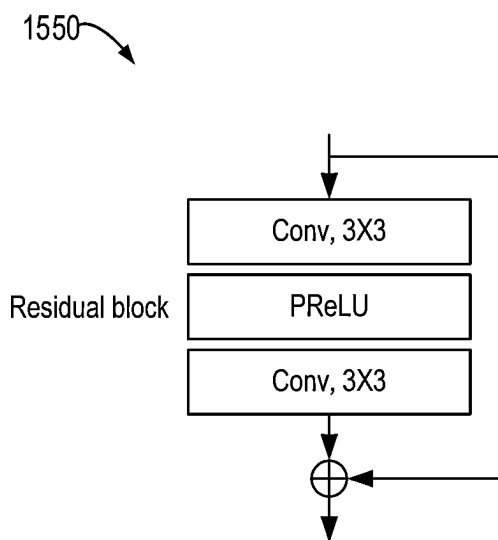


Fig. 15B

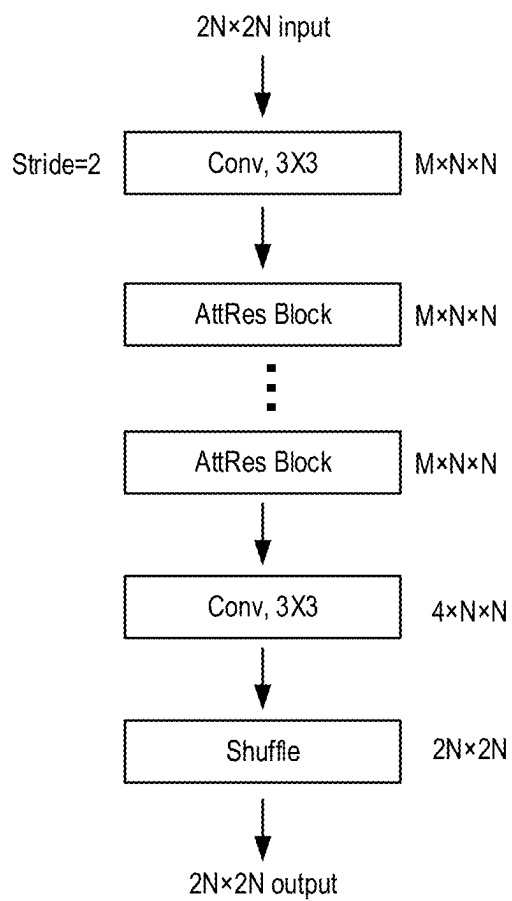


Fig. 16A

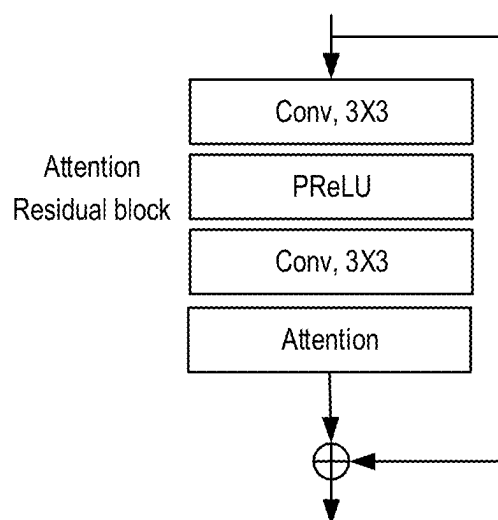


Fig. 16B

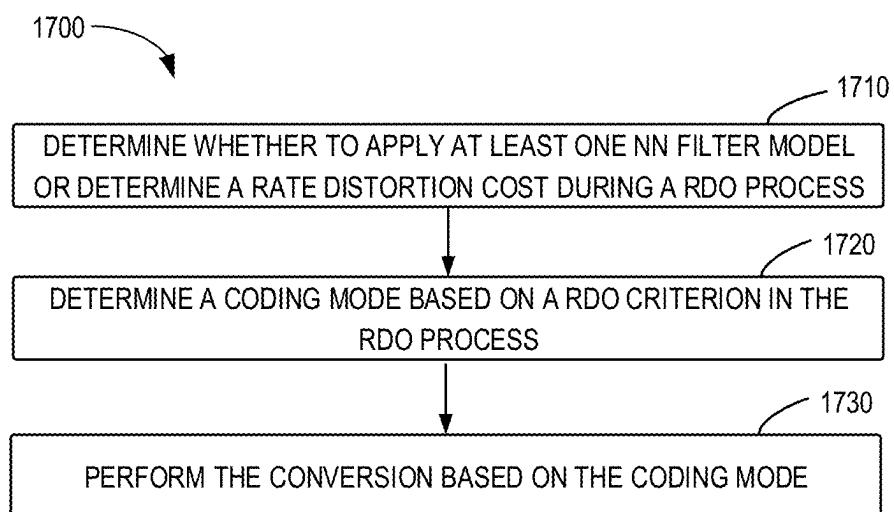


Fig. 17

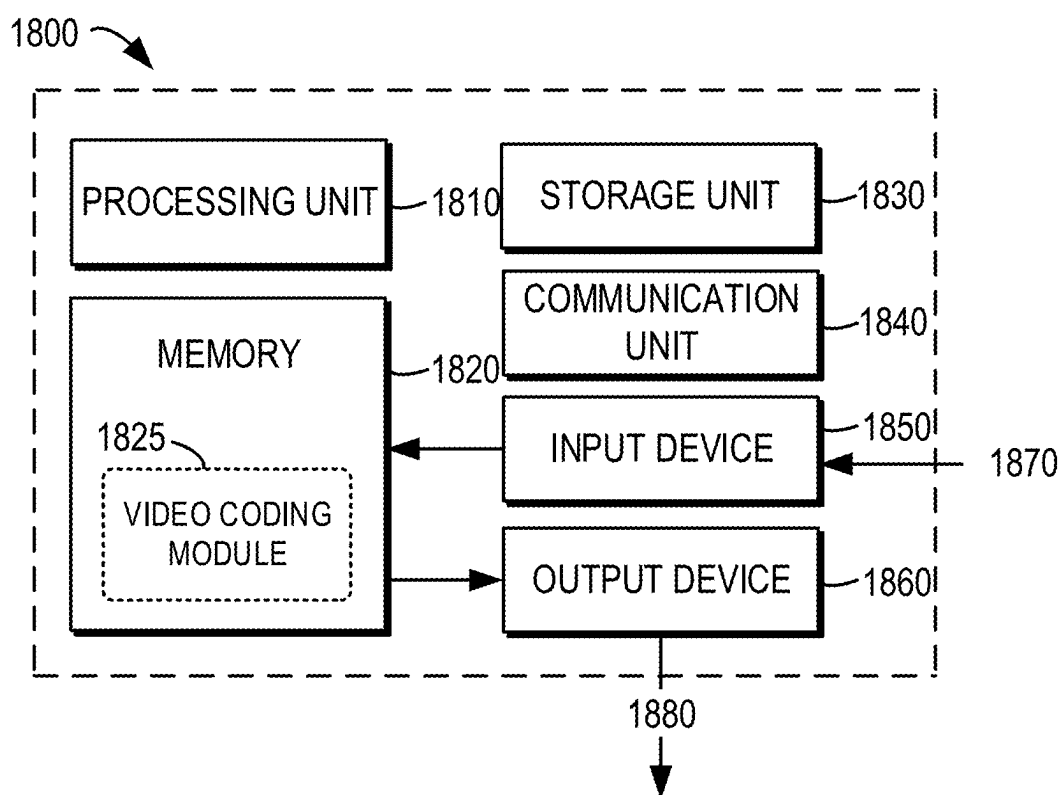


Fig. 18

METHOD, APPARATUS, AND MEDIUM FOR VIDEO PROCESSING

CROSS REFERENCE

[0001] This application is a continuation of International Application No. PCT/CN2023/124366, filed on Oct. 12, 2023, which claims the benefit of International Application No. PCT/CN2022/125228, filed on Oct. 13, 2022. The entire contents of these applications are hereby incorporated by reference in their entireties.

FIELDS

[0002] Embodiments of the present disclosure relates generally to video processing techniques, and more particularly, to neural network (NN) in-loop filtering based content-adaptive rate distortion optimization for image/video coding.

BACKGROUND

[0003] In nowadays, digital video capabilities are being applied in various aspects of peoples' lives. Multiple types of video compression technologies, such as MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4 Part 10 Advanced Video Coding (AVC), ITU-T H.265 high efficiency video coding (HEVC) standard, versatile video coding (VVC) standard, have been proposed for video encoding/decoding. However, coding efficiency of video coding techniques is generally expected to be further improved.

SUMMARY

[0004] Embodiments of the present disclosure provide a solution for video processing.

[0005] In a first aspect, a method for video processing is proposed. The method comprises: determining, for a conversion between a video unit of a video and a bitstream of the video unit, whether to apply at least one neural network (NN) filter model or determine a rate distortion cost during a rate distortion optimization (RDO) process of the video unit based on at least one of: a distortion without NN filter model, a distortion with n-th NN filter model, a combination of distortions of a plurality of NN filter models, or coding statistics of the video unit, and wherein n is an integer number; determining a coding mode of the video unit based on a rate distortion optimization (RDO) criterion in the RDO process; and performing the conversion based on the coding mode. In this way, the impact of reduce distortion due to NN filter is taken into consideration during the RDO process, thereby improving coding performances.

[0006] In a second aspect, an apparatus for video processing is proposed. The apparatus comprises a processor and a non-transitory memory with instructions thereon. The instructions upon execution by the processor, cause the processor to perform a method in accordance with the first aspect of the present disclosure.

[0007] In a third aspect, a non-transitory computer-readable storage medium is proposed. The non-transitory computer-readable storage medium stores instructions that cause a processor to perform a method in accordance with the first aspect of the present disclosure.

[0008] In a fourth aspect, another non-transitory computer-readable recording medium is proposed. The non-transitory computer-readable recording medium stores a bitstream of a video which is generated by a method

performed by an apparatus for video processing. The method comprises: determining whether to apply at least one neural network (NN) filter model or determine a rate distortion cost during a rate distortion optimization (RDO) process of a video unit of the video based on at least one of: a distortion without NN filter model, a distortion with n-th NN filter model, a combination of distortions of a plurality of NN filter models, or coding statistics of the video unit, and wherein n is an integer number; determining a coding mode of the video unit based on a rate distortion optimization (RDO) criterion in the RDO process; and generating the bitstream based on the coding mode.

[0009] In a fifth aspect, a method for storing a bitstream of a video is proposed. The method comprises: determining whether to apply at least one neural network (NN) filter model or determine a rate distortion cost during a rate distortion optimization (RDO) process of a video unit of the video based on at least one of: a distortion without NN filter model, a distortion with n-th NN filter model, a combination of distortions of a plurality of NN filter models, or coding statistics of the video unit, and wherein n is an integer number; determining a coding mode of the video unit based on a rate distortion optimization (RDO) criterion in the RDO process; generating the bitstream based on the coding mode; and storing the bitstream in a non-transitory computer-readable recording medium.

[0010] This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used to limit the scope of the claimed subject matter.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] Through the following detailed description with reference to the accompanying drawings, the above and other objectives, features, and advantages of example embodiments of the present disclosure will become more apparent. In the example embodiments of the present disclosure, the same reference numerals usually refer to the same components.

[0012] FIG. 1 illustrates a block diagram that illustrates an example video coding system, in accordance with some embodiments of the present disclosure;

[0013] FIG. 2 illustrates a block diagram that illustrates a first example video encoder, in accordance with some embodiments of the present disclosure;

[0014] FIG. 3 illustrates a block diagram that illustrates an example video decoder, in accordance with some embodiments of the present disclosure;

[0015] FIG. 4 illustrates an example diagram showing an example of raster-scan slice partitioning of a picture;

[0016] FIG. 5 illustrates an example diagram showing an example of rectangular slice partitioning of a picture;

[0017] FIG. 6 illustrates an example diagram showing an example of a picture partitioned into tiles, bricks, and rectangular slices;

[0018] FIG. 7A illustrates an example diagram showing CTBs crossing the bottom picture border;

[0019] FIG. 7B illustrates an example diagram showing CTBs crossing the right picture border;

[0020] FIG. 7C illustrates an example diagram showing CTBs crossing the right bottom picture border;

[0021] FIG. 8 illustrates an example diagram showing an example of encoder block diagram;

[0022] FIG. 9 illustrates an example diagram showing an illustration of picture samples and horizontal and vertical block boundaries on the 8×8 grid, and the nonoverlapping blocks of the 8×8 samples;

[0023] FIG. 10 illustrates an example diagram showing pixels involved in filter on/off decision and strong/weak filter selection;

[0024] FIGS. 11A-11D illustrate example diagrams showing four 1-D directional patterns for EO sample classification;

[0025] FIGS. 12A-12C illustrate example diagrams showing examples of GALF filter shapes;

[0026] FIGS. 13A-13C illustrate example diagrams showing examples of relative coordinator for the 5×5 diamond filter support;

[0027] FIG. 14 illustrates an example diagram showing examples of relative coordinates for the 5×5 diamond filter support;

[0028] FIG. 15A illustrates an example diagram showing architecture of the proposed CNN filter;

[0029] FIG. 15B illustrates an example diagram showing a construction of ResBlock (residual block) in the CNN filter;

[0030] FIG. 16A illustrates an example diagram showing architecture of the proposed CNN filter;

[0031] FIG. 16B illustrates an example diagram showing a construction of Attention Residual Block in FIG. 16A;

[0032] FIG. 17 illustrates a flowchart of a method for video processing in accordance with embodiments of the present disclosure; and

[0033] FIG. 18 illustrates a block diagram of a computing device in which various embodiments of the present disclosure can be implemented.

[0034] Throughout the drawings, the same or similar reference numerals usually refer to the same or similar elements.

DETAILED DESCRIPTION

[0035] Principle of the present disclosure will now be described with reference to some embodiments. It is to be understood that these embodiments are described only for the purpose of illustration and help those skilled in the art to understand and implement the present disclosure, without suggesting any limitation as to the scope of the disclosure. The disclosure described herein can be implemented in various manners other than the ones described below.

[0036] In the following description and claims, unless defined otherwise, all technical and scientific terms used herein have the same meaning as commonly understood by one of ordinary skills in the art to which this disclosure belongs.

[0037] References in the present disclosure to “one embodiment,” “an embodiment,” “an example embodiment,” and the like indicate that the embodiment described may include a particular feature, structure, or characteristic, but it is not necessary that every embodiment includes the particular feature, structure, or characteristic. Moreover, such phrases are not necessarily referring to the same embodiment. Further, when a particular feature, structure, or characteristic is described in connection with an example embodiment, it is submitted that it is within the knowledge of one skilled in the art to affect such feature, structure, or

characteristic in connection with other embodiments whether or not explicitly described.

[0038] It shall be understood that although the terms “first” and “second” etc. may be used herein to describe various elements, these elements should not be limited by these terms. These terms are only used to distinguish one element from another. For example, a first element could be termed a second element, and similarly, a second element could be termed a first element, without departing from the scope of example embodiments. As used herein, the term “and/or” includes any and all combinations of one or more of the listed terms.

[0039] The terminology used herein is for the purpose of describing particular embodiments only and is not intended to be limiting of example embodiments. As used herein, the singular forms “a,” “an” and “the” are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will be further understood that the terms “comprises,” “comprising,” “has,” “having,” “includes” and/or “including,” when used herein, specify the presence of stated features, elements, and/or components etc., but do not preclude the presence or addition of one or more other features, elements, components and/or combinations thereof.

Example Environment

[0040] FIG. 1 is a block diagram that illustrates an example video coding system 100 that may utilize the techniques of this disclosure. As shown, the video coding system 100 may include a source device 110 and a destination device 120. The source device 110 can be also referred to as a video encoding device, and the destination device 120 can be also referred to as a video decoding device. In operation, the source device 110 can be configured to generate encoded video data and the destination device 120 can be configured to decode the encoded video data generated by the source device 110. The source device 110 may include a video source 112, a video encoder 114, and an input/output (I/O) interface 116.

[0041] The video source 112 may include a source such as a video capture device. Examples of the video capture device include, but are not limited to, an interface to receive video data from a video content provider, a computer graphics system for generating video data, and/or a combination thereof.

[0042] The video data may comprise one or more pictures. The video encoder 114 encodes the video data from the video source 112 to generate a bitstream. The bitstream may include a sequence of bits that form a coded representation of the video data. The bitstream may include coded pictures and associated data. The coded picture is a coded representation of a picture. The associated data may include sequence parameter sets, picture parameter sets, and other syntax structures. The I/O interface 116 may include a modulator/demodulator and/or a transmitter. The encoded video data may be transmitted directly to destination device 120 via the I/O interface 116 through the network 130A. The encoded video data may also be stored onto a storage medium/server 130B for access by destination device 120.

[0043] The destination device 120 may include an I/O interface 126, a video decoder 124, and a display device 122. The I/O interface 126 may include a receiver and/or a modem. The I/O interface 126 may acquire encoded video data from the source device 110 or the storage medium/

server 130B. The video decoder 124 may decode the encoded video data. The display device 122 may display the decoded video data to a user. The display device 122 may be integrated with the destination device 120, or may be external to the destination device 120 which is configured to interface with an external display device.

[0044] The video encoder 114 and the video decoder 124 may operate according to a video compression standard, such as the High Efficiency Video Coding (HEVC) standard, Versatile Video Coding (VVC) standard and other current and/or further standards.

[0045] FIG. 2 is a block diagram illustrating an example of a video encoder 200, which may be an example of the video encoder 114 in the system 100 illustrated in FIG. 1, in accordance with some embodiments of the present disclosure.

[0046] The video encoder 200 may be configured to implement any or all of the techniques of this disclosure. In the example of FIG. 2, the video encoder 200 includes a plurality of functional components. The techniques described in this disclosure may be shared among the various components of the video encoder 200. In some examples, a processor may be configured to perform any or all of the techniques described in this disclosure.

[0047] In some embodiments, the video encoder 200 may include a partition unit 201, a predication unit 202 which may include a mode select unit 203, a motion estimation unit 204, a motion compensation unit 205 and an intra-prediction unit 206, a residual generation unit 207, a transform unit 208, a quantization unit 209, an inverse quantization unit 210, an inverse transform unit 211, a reconstruction unit 212, a buffer 213, and an entropy encoding unit 214.

[0048] In other examples, the video encoder 200 may include more, fewer, or different functional components. In an example, the predication unit 202 may include an intra block copy (IBC) unit. The IBC unit may perform predication in an IBC mode in which at least one reference picture is a picture where the current video block is located.

[0049] Furthermore, although some components, such as the motion estimation unit 204 and the motion compensation unit 205, may be integrated, but are represented in the example of FIG. 2 separately for purposes of explanation.

[0050] The partition unit 201 may partition a picture into one or more video blocks. The video encoder 200 and the video decoder 300 may support various video block sizes.

[0051] The mode select unit 203 may select one of the coding modes, intra or inter, e.g., based on error results, and provide the resulting intra-coded or inter-coded block to a residual generation unit 207 to generate residual block data and to a reconstruction unit 212 to reconstruct the encoded block for use as a reference picture. In some examples, the mode select unit 203 may select a combination of intra and inter predication (CIIP) mode in which the predication is based on an inter predication signal and an intra predication signal. The mode select unit 203 may also select a resolution for a motion vector (e.g., a sub-pixel or integer pixel precision) for the block in the case of inter-predication.

[0052] To perform inter prediction on a current video block, the motion estimation unit 204 may generate motion information for the current video block by comparing one or more reference frames from buffer 213 to the current video block. The motion compensation unit 205 may determine a predicted video block for the current video block based on

the motion information and decoded samples of pictures from the buffer 213 other than the picture associated with the current video block.

[0053] The motion estimation unit 204 and the motion compensation unit 205 may perform different operations for a current video block, for example, depending on whether the current video block is in an I-slice, a P-slice, or a B-slice. As used herein, an “I-slice” may refer to a portion of a picture composed of macroblocks, all of which are based upon macroblocks within the same picture. Further, as used herein, in some aspects, “P-slices” and “B-slices” may refer to portions of a picture composed of macroblocks that are not dependent on macroblocks in the same picture.

[0054] In some examples, the motion estimation unit 204 may perform uni-directional prediction for the current video block, and the motion estimation unit 204 may search reference pictures of list 0 or list 1 for a reference video block for the current video block. The motion estimation unit 204 may then generate a reference index that indicates the reference picture in list 0 or list 1 that contains the reference video block and a motion vector that indicates a spatial displacement between the current video block and the reference video block. The motion estimation unit 204 may output the reference index, a prediction direction indicator, and the motion vector as the motion information of the current video block. The motion compensation unit 205 may generate the predicted video block of the current video block based on the reference video block indicated by the motion information of the current video block.

[0055] Alternatively, in other examples, the motion estimation unit 204 may perform bi-directional prediction for the current video block. The motion estimation unit 204 may search the reference pictures in list 0 for a reference video block for the current video block and may also search the reference pictures in list 1 for another reference video block for the current video block. The motion estimation unit 204 may then generate reference indexes that indicate the reference pictures in list 0 and list 1 containing the reference video blocks and motion vectors that indicate spatial displacements between the reference video blocks and the current video block. The motion estimation unit 204 may output the reference indexes and the motion vectors of the current video block as the motion information of the current video block. The motion compensation unit 205 may generate the predicted video block of the current video block based on the reference video blocks indicated by the motion information of the current video block.

[0056] In some examples, the motion estimation unit 204 may output a full set of motion information for decoding processing of a decoder. Alternatively, in some embodiments, the motion estimation unit 204 may signal the motion information of the current video block with reference to the motion information of another video block. For example, the motion estimation unit 204 may determine that the motion information of the current video block is sufficiently similar to the motion information of a neighboring video block.

[0057] In one example, the motion estimation unit 204 may indicate, in a syntax structure associated with the current video block, a value that indicates to the video decoder 300 that the current video block has the same motion information as the another video block.

[0058] In another example, the motion estimation unit 204 may identify, in a syntax structure associated with the current video block, another video block and a motion vector

difference (MVD). The motion vector difference indicates a difference between the motion vector of the current video block and the motion vector of the indicated video block. The video decoder 300 may use the motion vector of the indicated video block and the motion vector difference to determine the motion vector of the current video block.

[0059] As discussed above, video encoder 200 may predictively signal the motion vector. Two examples of predictive signaling techniques that may be implemented by video encoder 200 include advanced motion vector predication (AMVP) and merge mode signaling.

[0060] The intra prediction unit 206 may perform intra prediction on the current video block. When the intra prediction unit 206 performs intra prediction on the current video block, the intra prediction unit 206 may generate prediction data for the current video block based on decoded samples of other video blocks in the same picture. The prediction data for the current video block may include a predicted video block and various syntax elements.

[0061] The residual generation unit 207 may generate residual data for the current video block by subtracting (e.g., indicated by the minus sign) the predicted video block(s) of the current video block from the current video block. The residual data of the current video block may include residual video blocks that correspond to different sample components of the samples in the current video block.

[0062] In other examples, there may be no residual data for the current video block for the current video block, for example in a skip mode, and the residual generation unit 207 may not perform the subtracting operation.

[0063] The transform processing unit 208 may generate one or more transform coefficient video blocks for the current video block by applying one or more transforms to a residual video block associated with the current video block.

[0064] After the transform processing unit 208 generates a transform coefficient video block associated with the current video block, the quantization unit 209 may quantize the transform coefficient video block associated with the current video block based on one or more quantization parameter (QP) values associated with the current video block.

[0065] The inverse quantization unit 210 and the inverse transform unit 211 may apply inverse quantization and inverse transforms to the transform coefficient video block, respectively, to reconstruct a residual video block from the transform coefficient video block. The reconstruction unit 212 may add the reconstructed residual video block to corresponding samples from one or more predicted video blocks generated by the predication unit 202 to produce a reconstructed video block associated with the current video block for storage in the buffer 213.

[0066] After the reconstruction unit 212 reconstructs the video block, loop filtering operation may be performed to reduce video blocking artifacts in the video block.

[0067] The entropy encoding unit 214 may receive data from other functional components of the video encoder 200. When the entropy encoding unit 214 receives the data, the entropy encoding unit 214 may perform one or more entropy encoding operations to generate entropy encoded data and output a bitstream that includes the entropy encoded data.

[0068] FIG. 3 is a block diagram illustrating an example of a video decoder 300, which may be an example of the video

decoder 124 in the system 100 illustrated in FIG. 1, in accordance with some embodiments of the present disclosure.

[0069] The video decoder 300 may be configured to perform any or all of the techniques of this disclosure. In the example of FIG. 3, the video decoder 300 includes a plurality of functional components. The techniques described in this disclosure may be shared among the various components of the video decoder 300. In some examples, a processor may be configured to perform any or all of the techniques described in this disclosure.

[0070] In the example of FIG. 3, the video decoder 300 includes an entropy decoding unit 301, a motion compensation unit 302, an intra prediction unit 303, an inverse quantization unit 304, an inverse transformation unit 305, and a reconstruction unit 306 and a buffer 307. The video decoder 300 may, in some examples, perform a decoding pass generally reciprocal to the encoding pass described with respect to video encoder 200.

[0071] The entropy decoding unit 301 may retrieve an encoded bitstream. The encoded bitstream may include entropy coded video data (e.g., encoded blocks of video data). The entropy decoding unit 301 may decode the entropy coded video data, and from the entropy decoded video data, the motion compensation unit 302 may determine motion information including motion vectors, motion vector precision, reference picture list indexes, and other motion information. The motion compensation unit 302 may, for example, determine such information by performing the AMVP and merge mode. AMVP is used, including derivation of several most probable candidates based on data from adjacent PBs and the reference picture. Motion information typically includes the horizontal and vertical motion vector displacement values, one or two reference picture indices, and, in the case of prediction regions in B slices, an identification of which reference picture list is associated with each index. As used herein, in some aspects, a “merge mode” may refer to deriving the motion information from spatially or temporally neighboring blocks.

[0072] The motion compensation unit 302 may produce motion compensated blocks, possibly performing interpolation based on interpolation filters. Identifiers for interpolation filters to be used with sub-pixel precision may be included in the syntax elements.

[0073] The motion compensation unit 302 may use the interpolation filters as used by the video encoder 200 during encoding of the video block to calculate interpolated values for sub-integer pixels of a reference block. The motion compensation unit 302 may determine the interpolation filters used by the video encoder 200 according to the received syntax information and use the interpolation filters to produce predictive blocks.

[0074] The motion compensation unit 302 may use at least part of the syntax information to determine sizes of blocks used to encode frame(s) and/or slice(s) of the encoded video sequence, partition information that describes how each macroblock of a picture of the encoded video sequence is partitioned, modes indicating how each partition is encoded, one or more reference frames (and reference frame lists) for each inter-encoded block, and other information to decode the encoded video sequence. As used herein, in some aspects, a “slice” may refer to a data structure that can be decoded independently from other slices of the same picture,

in terms of entropy coding, signal prediction, and residual signal reconstruction. A slice can either be an entire picture or a region of a picture.

[0075] The intra prediction unit **303** may use intra prediction modes for example received in the bitstream to form a prediction block from spatially adjacent blocks. The inverse quantization unit **304** inverse quantizes, i.e., de-quantizes, the quantized video block coefficients provided in the bitstream and decoded by entropy decoding unit **301**. The inverse transform unit **305** applies an inverse transform.

[0076] The reconstruction unit **306** may obtain the decoded blocks, e.g., by summing the residual blocks with the corresponding prediction blocks generated by the motion compensation unit **302** or intra-prediction unit **303**. If desired, a deblocking filter may also be applied to filter the decoded blocks in order to remove blockiness artifacts. The decoded video blocks are then stored in the buffer **307**, which provides reference blocks for subsequent motion compensation/intra prediction and also produces decoded video for presentation on a display device.

[0077] Some exemplary embodiments of the present disclosure will be described in detailed hereinafter. It should be understood that section headings are used in the present document to facilitate ease of understanding and do not limit the embodiments disclosed in a section to only that section. Furthermore, while certain embodiments are described with reference to Versatile Video Coding or other specific video codecs, the disclosed techniques are applicable to other video coding technologies also. Furthermore, while some embodiments describe video coding steps in detail, it will be understood that corresponding steps decoding that undo the coding will be implemented by a decoder. Furthermore, the term video processing encompasses video coding or compression, video decoding or decompression and video transcoding in which video pixels are represented from one compressed format into another compressed format or at a different compressed bitrate.

1. Brief Summary

[0078] This disclosure is related to video coding technologies. Specifically, it is related to the loop filter in image/video coding. It may be applied to the existing video coding standard like High-Efficiency Video Coding (HEVC), Versatile Video Coding (VVC), or the standard (e.g., AVS3) to be finalized. It may be also applicable to future video coding standards or video codec or being used as post-processing method which is out of encoding/decoding process.

2. Introduction

[0079] Video coding standards have evolved primarily through the development of the well-known ITU-T and ISO/IEC standards. The ITU-T produced H.261 and H.263, ISO/IEC produced MPEG-1 and MPEG-4 Visual, and the two organizations jointly produced the H.262/MPEG-2 Video and H.264/MPEG-4 Advanced Video Coding (AVC) and H.265/HEVC standards. Since H.262, the video coding standards are based on the hybrid video coding structure where temporal prediction plus transform coding are utilized. To explore the future video coding technologies beyond HEVC, Joint Video Exploration Team (JVET) was founded by VCEG and MPEG jointly in 2015. Since then, many new methods have been adopted by JVET and put into the reference software named Joint Exploration Model

(JEM). In April 2018, the Joint Video Expert Team (JVET) between VCEG (Q6/16) and ISO/IEC JTC1 SC29/WG11 (MPEG) was created to work on the VVC standard targeting at 50% bitrate reduction compared to HEVC. VVC version 1 was finalized in July 2020.

[0080] The latest version of VVC draft, i.e., Versatile Video Coding (Draft 10) could be found at: http://phenix.it-sudparis.eu/jvet/doc_end_user/current_document.php?id=10399.

[0081] The latest reference software of VVC, named VTM, could be found at: https://vcgit.hhi.fraunhofer.de/jvet/VVCSOFTWARE_VTM/-/tags/VTM-10.0.

2.1. Color Space and Chroma Subsampling

[0082] Color space, also known as the color model (or color system), is an abstract mathematical model which simply describes the range of colors as tuples of numbers, typically as 3 or 4 values or color components (e.g. RGB). Basically speaking, color space is an elaboration of the coordinate system and sub-space. For video compression, the most frequently used color spaces are YCbCr and RGB.

[0083] YCbCr, Y'CbCr, or Y Pb/Cb Pr/Cr, also written as YCBCR or Y'CBCR, is a family of color spaces used as a part of the color image pipeline in video and digital photography systems. Y' is the luma component and Cb and Cr are the blue-difference and red-difference chroma components. Y' (with prime) is distinguished from Y, which is luminance, meaning that light intensity is nonlinearly encoded based on gamma corrected RGB primaries.

[0084] Chroma subsampling is the practice of encoding images by implementing less resolution for chroma information than for luma information, taking advantage of the human visual system's lower acuity for color differences than for luminance.

2.1.1. 4:4:4

[0085] Each of the three Y'CbCr components have the same sample rate, thus there is no chroma subsampling. This scheme is sometimes used in high-end film scanners and cinematic post production.

2.1.2. 4:2:2

[0086] The two chroma components are sampled at half the sample rate of luma: the horizontal chroma resolution is halved. This reduces the bandwidth of an uncompressed video signal by one-third with little to no visual difference.

2.1.3. 4:2:0

[0087] In 4:2:0, the horizontal sampling is doubled compared to 4:1:1, but as the Cb and Cr channels are only sampled on each alternate line in this scheme, the vertical resolution is halved. The data rate is thus the same. Cb and Cr are each subsampled at a factor of 2 both horizontally and vertically. There are three variants of 4:2:0 schemes, having different horizontal and vertical siting.

[0088] In MPEG-2, Cb and Cr are cosited horizontally. Cb and Cr are sited between pixels in the vertical direction (sited interstitially).

[0089] In JPEG/JFIF, H.261, and MPEG-1, Cb and Cr are sited interstitially, halfway between alternate luma samples.

[0090] In 4:2:0 DV, Cb and Cr are co-sited in the horizontal direction. In the vertical direction, they are co-sited on alternating lines.

2.2. Definitions of Video Units

[0091] A picture is divided into one or more tile rows and one or more tile columns. A tile is a sequence of CTUs that covers a rectangular region of a picture.

[0092] A tile is divided into one or more bricks, each of which consisting of a number of CTU rows within the tile.

[0093] A tile that is not partitioned into multiple bricks is also referred to as a brick. However, a brick that is a true subset of a tile is not referred to as a tile.

[0094] A slice either contains a number of tiles of a picture or a number of bricks of a tile.

[0095] Two modes of slices are supported, namely the raster-scan slice mode and the rectangular slice mode. In the raster-scan slice mode, a slice contains a sequence of tiles in a tile raster scan of a picture. In the rectangular slice mode, a slice contains a number of bricks of a picture that collectively form a rectangular region of the picture. The bricks within a rectangular slice are in the order of brick raster scan of the slice. FIG. 4 illustrates an example diagram 400 showing an example of raster-scan slice partitioning of a picture. In FIG. 4, the picture is divided into 12 tiles and 3 raster-scan slices. The picture in FIG. 4 with 18 by 12 luma CTUs is partitioned into 12 tiles and 3 raster-scan slices (informative).

[0096] FIG. 5 illustrates an example diagram 500 showing an example of rectangular slice partitioning of a picture. In FIG. 5, the picture is divided into 24 tiles (6 tile columns and 4 tile rows) and 9 rectangular slices. The picture in FIG. 5 with 18 by 12 luma CTUs is partitioned into 24 tiles and 9 rectangular slices (informative). FIG. 6 illustrates an example diagram 600 showing an example of a picture partitioned into tiles, bricks, and rectangular slices. In FIG. 6, the picture is divided into 4 tiles (2 tile columns and 2 tile rows), 11 bricks (the top-left tile contains 1 brick, the top-right tile contains 5 bricks, the bottom-left tile contains 2 bricks, and the bottom-right tile contain 3 bricks), and 4 rectangular slices. The picture in FIG. 6 is partitioned into 4 tiles, 11 bricks, and 4 rectangular slices (informative).

2.2.1. CTU/CTB Sizes

[0097] In VVC, the CTU size, signaled in SPS by the syntax element `log_2_ctu_size_minus2`, could be as small as 4x4.

7.3.2.3 Sequence parameter set RBSP syntax	
	De- scrip- tor
<code>seq_parameter_set_rbsp() {</code>	
<code>sps_decoding_parameter_set_id</code>	u(4)
<code>sps_video_parameter_set_id</code>	u(4)
<code>sps_max_sub_layers_minus1</code>	u(3)
<code>sps_reserved_zero_5bits</code>	u(5)
<code>profile_tier_level(sps_max_sub_layers_minus1)</code>	
<code>gra_enabled_flag</code>	u(1)
<code>sps_seq_parameter_set_id</code>	ue(v)
<code>chroma_format_idc</code>	ue(v)
<code>if(chroma_format_idc == 3)</code>	
<code>separate_colour_plane_flag</code>	u(1)
<code>pic_width_in_luma_samples</code>	ue(v)
<code>pic_height_in_luma_samples</code>	ue(v)
<code>conformance_window_flag</code>	u(1)
<code>if(conformance_window_flag) {</code>	
<code>conf_win_left_offset</code>	ue(v)
<code>conf_win_right_offset</code>	ue(v)

-continued

7.3.2.3 Sequence parameter set RBSP syntax	
	De- scrip- tor
<code>conf_win_top_offset</code>	ue(v)
<code>conf_win_bottom_offset</code>	ue(v)
<code>}</code>	
<code>bit_depth_luma_minus8</code>	ue(v)
<code>bit_depth_chroma_minus8</code>	ue(v)
<code>log2_max_pic_order_cnt_lsb_minus4</code>	ue(v)
<code>sps_sub_layer_ordering_info_present_flag</code>	u(1)
<code>for(i = (sps_sub_layer_ordering_info_present_flag ? 0 :</code>	
<code>sps_max_sub_layers_minus1);</code>	
<code>i <= sps_max_sub_layers_minus1; i++) {</code>	
<code>sps_max_dec_pic_buffering_minus1[i]</code>	ue(v)
<code>sps_max_num_reorder_pics[i]</code>	ue(v)
<code>sps_max_latency_increase_plus1[i]</code>	ue(v)
<code>}</code>	
<code>long_term_ref_pics_flag</code>	u(1)
<code>sps_idr_rpl_present_flag</code>	u(1)
<code>rpl1_same_as_rpl0_flag</code>	u(1)
<code>for(i = 0; i < !rpl1_same_as_rpl0_flag ? 2 : 1; i++) {</code>	
<code>num_ref_pic_lists_in_sps[i]</code>	ue(v)
<code>for(j = 0; j < num_ref_pic_lists_in_sps[i]; j++)</code>	
<code>ref_pic_list_struct(i, j)</code>	
<code>}</code>	
<code>qtbtt_dual_tree_intra_flag</code>	u(1)
<code>log2_ctu_size_minus2</code>	ue(v)
<code>log2_min_luma_coding_block_size_minus2</code>	ue(v)
<code>partition_constraints_override_enabled_flag</code>	u(1)
<code>sps_log2_diff_min_qt_min_cb_intra_slice_luma</code>	ue(v)
<code>sps_log2_diff_min_qt_min_cb_inter_slice</code>	ue(v)
<code>sps_max_mtt_hierarchy_depth_inter_slice</code>	ue(v)
<code>sps_max_mtt_hierarchy_depth_intra_slice_luma</code>	ue(v)
<code>if(sps_max_mtt_hierarchy_depth_intra_slice_luma != 0) {</code>	
<code>sps_log2_diff_max_bt_min_qt_intra_slice_luma</code>	ue(v)
<code>sps_log2_diff_max_tt_min_qt_intra_slice_luma</code>	ue(v)
<code>}</code>	
<code>if(sps_max_mtt_hierarchy_depth_inter_slices != 0) {</code>	
<code>sps_log2_diff_max_bt_min_qt_inter_slice</code>	ue(v)
<code>sps_log2_diff_max_tt_min_qt_inter_slice</code>	ue(v)
<code>}</code>	
<code>if(qtbtt_dual_tree_intra_flag) {</code>	
<code>sps_log2_diff_min_qt_min_cb_intra_slice_chroma</code>	ue(v)
<code>sps_max_mtt_hierarchy_depth_intra_slice_chroma</code>	ue(v)
<code>if(sps_max_mtt_hierarchy_depth_intra_slice_chroma</code>	
<code>!= 0) {</code>	
<code>sps_log2_diff_max_bt_min_qt_intra_slice_chroma</code>	ue(v)
<code>sps_log2_diff_max_tt_min_qt_intra_slice_chroma</code>	ue(v)
<code>}</code>	
<code>}</code>	
<code>...</code>	
<code>rbsp_trailing_bits()</code>	
<code>}</code>	

[0098] `log_2_ctu_size_minus2` plus 2 specifies the luma coding tree block size of each CTU.

[0099] `log_min_luma_coding_block_size_minus2` plus 2 specifies the minimum luma coding block size.

[0100] The variables `CtbLog_2Size_Y`, `CtbSizeY`, `MinCbLog_2SizeY`, `MinCbSizeY`, `MinTbLog_2Size_Y`, `MaxTbLog_2SizeY`, `MinTbSizeY`, `MaxTbSize_Y`, `PicWidthInCtbsY`, `PicHeightInCtbsY`, `PicSizeInCtbsY`, `PicWidthInMinCbsY`, `PicHeightInMinCbsY`, `PicSizeInMinCbsY`, `PicSizeInSamplesY`, `PicWidthInSamplesC` and `PicHeightInSamplesC` are derived as follows:

$$CtbLog2SizeY = \log_2 \text{ctu_size_minus2} + 2 \quad (7-9)$$

$$CtbSizeY = 1 \ll CtbLog2SizeY \quad (7-10)$$

$$\text{MinCbLog2SizeY} = \log_2 \text{min_luma_coding_block_size_minus2} + 2 \quad (7-11)$$

$$\text{MinCbSizeY} = 1 \ll \text{MinCbLog2SizeY} \quad (7-12)$$

$$\text{MinTbLog2SizeY} = 2 \quad (7-13)$$

$$\text{MaxTbLog2SizeY} = 6 \quad (7-14)$$

$$\text{MinTbSizeY} = 1 \ll \text{MinTbLog2SizeY} \quad (7-15)$$

$$\text{MaxTbSizeY} = 1 \ll \text{MaxTbLog2SizeY} \quad (7-16)$$

$$\text{PicWidthInCtbsY} = \text{Ceil}(\text{pic_width_in_luma_samples} \div \text{CtbSizeY}) \quad (7-17)$$

$$\text{PicHeightInCtbsY} = \text{Ceil}(\text{pic_height_in_luma_samples} \div \text{CtbSizeY}) \quad (7-18)$$

$$\text{PicSizeInCtbsY} = \text{PicWidthInCtbsY} * \text{PicHeightInCtbsY} \quad (7-19)$$

$$\text{PicWidthInMinCbsY} = \text{pic_width_in_luma_samples} / \text{MinCbSizeY} \quad (7-20)$$

$$\text{PicHeightInMinCbsY} = \text{pic_height_in_luma_samples} / \text{MinCbSizeY} \quad (7-21)$$

$$\text{PicSizeInMinCbsY} = \text{PicWidthInMinCbsY} * \text{PicHeightInMinCbsY} \quad (7-22)$$

$$\text{PicSizeInSamplesY} = \text{pic_width_in_luma_samples} * \text{pic_height_in_luma_samples} \quad (7-23)$$

$$\text{PicWidthInSamplesC} = \text{pic_width_in_luma_samples} / \text{SubWidthC} \quad (7-24)$$

$$\text{PicHeightInSamplesC} = \text{pic_height_in_luma_samples} / \text{SubHeightC} \quad (7-25)$$

2.2.2. CTUs in a Picture

[0101] Suppose the CTB/LCU size indicated by $M \times N$ (typically M is equal to N , as defined in HEVC/VVC), and for a CTB located at picture (or tile or slice or other kinds of types, picture border is taken as an example) border, $K \times L$ samples are within picture border where either $K < M$ or $L < N$. FIG. 7A illustrate an example diagram **700** showing CTBs crossing the bottom picture border, in which $K = M$, $L < N$. FIG. 7B illustrates an example diagram **720** showing CTBs crossing the right picture border, in which $K < M$, $L = N$. FIG. 7C illustrates an example diagram **740** showing CTBs crossing the right bottom picture border, in which $K < M$, $L < N$. For those CTBs as depicted in FIGS. 7A-7C, the CTB size is still equal to $M \times N$, however, the bottom boundary/right boundary of the CTB is outside the picture.

2.3. Coding Flow of a Typical Video Codec

[0102] FIG. 8 illustrates an example diagram **800** showing an example of encoder block diagram of VVC, which contains three in-loop filtering blocks: deblocking filter (DF) **805**, sample adaptive offset (SAO) **806** and ALF **807**. Unlike DF **805**, which uses predefined filters, SAO **806** and ALF **807** utilize the original samples of the current picture to reduce the mean square errors between the original samples and the reconstructed samples by adding an offset and by applying a finite impulse response (FIR) filter, respectively, with coded side information signaling the offsets and filter coefficients. ALF **807** is located at the last processing stage

of each picture and can be regarded as a tool trying to catch and fix artifacts created by the previous stages.

2.4. Deblocking Filter (DB)

The Input of DB is the Reconstructed Samples Before In-Loop Filters.

[0103] The vertical edges in a picture are filtered first. Then the horizontal edges in a picture are filtered with samples modified by the vertical edge filtering process as input. The vertical and horizontal edges in the CTBs of each CTU are processed separately on a coding unit basis. The vertical edges of the coding blocks in a coding unit are filtered starting with the edge on the left-hand side of the coding blocks proceeding through the edges towards the right-hand side of the coding blocks in their geometrical order. The horizontal edges of the coding blocks in a coding unit are filtered starting with the edge on the top of the coding blocks proceeding through the edges towards the bottom of the coding blocks in their geometrical order.

[0104] FIG. 9 illustrates an example diagram **900** showing an illustration of picture samples and horizontal and vertical block boundaries on the 8×8 grid, and the nonoverlapping blocks of the 8×8 samples, which can be deblocked in parallel.

2.4.1. Boundary Decision

[0105] Filtering is applied to 8×8 block boundaries. In addition, it must be a transform block boundary or a coding subblock boundary (e.g., due to usage of Affine motion prediction, ATMVP). For those which are not such boundaries, filter is disabled.

2.4.2. Boundary Strength Calculation

[0106] For a transform block boundary/coding subblock boundary, if it is located in the 8×8 grid, it may be filtered and the setting of $bs[xD_i][yD_j]$ (where $[xD_i][yD_j]$ denotes the coordinate) for this edge is defined in Table 1 and Table 2, respectively.

TABLE 1

Boundary strength (when SPS IBC is disabled)				
Priority	Conditions	Y	U	V
5	At least one of the adjacent blocks is intra	2	2	2
4	TU boundary and at least one of the adjacent blocks has non-zero transform coefficients	1	1	1
3	Reference pictures or number of MVs (1 for uni-prediction, 2 for bi-prediction) of the adjacent blocks are different	1	N/A	N/A
2	Absolute difference between the motion vectors of same reference picture that belong to the adjacent blocks is greater than or equal to one integer luma sample	1	N/A	N/A
1	Otherwise	0	0	0

TABLE 2

Boundary strength (when SPS IBC is enabled)				
Priority	Conditions	Y	U	V
8	At least one of the adjacent blocks is intra	2	2	2
7	TU boundary and at least one of the adjacent blocks has non-zero transform coefficients	1	1	1
6	Prediction mode of adjacent blocks is different (e.g., one is IBC, one is inter)	1		
5	Both IBC and absolute difference between the motion vectors that belong to the adjacent blocks is greater than or equal to one integer luma sample	1	N/A	N/A
4	Reference pictures or number of MVs (1 for uni-prediction, 2 for bi-prediction) of the adjacent blocks are different	1	N/A	N/A
3	Absolute difference between the motion vectors of same reference picture that belong to the adjacent blocks is greater than or equal to one integer luma sample	1	N/A	N/A
1	Otherwise	0	0	0

2.4.3. Deblocking Decision for Luma Component

[0107] The deblocking decision process is described in this sub-section. FIG. 10 illustrates an example diagram 1000 showing pixels involved in filter on/off decision and strong/weak filter selection.

[0108] Wider-stronger luma filter is filters are used only if all the Condition1, Condition2 and Condition 3 are TRUE.

[0109] The condition 1 is the “large block condition”. This condition detects whether the samples at P-side and Q-side belong to large blocks, which are represented by the variable bSidePisLargeBlk and bSideQisLargeBlk respectively. The bSidePisLargeBlk and bSideQisLargeBlk are defined as follows.

$bSidePisLargeBlk = ((\text{edge type is vertical and } p_0 \text{ belongs to } CU \text{ with width} \geq 32) \parallel (\text{edge type is horizontal and } p_0 \text{ belongs to } CU \text{ with height} \geq 32)) ? \text{TRUE} : \text{FALSE}$

$bSideQisLargeBlk = ((\text{edge type is vertical and } q_0 \text{ belongs to } CU \text{ with width} \geq 32) \parallel (\text{edge type is horizontal and } q_0 \text{ belongs to } CU \text{ with height} \geq 32)) ? \text{TRUE} : \text{FALSE}$

[0110] Based on bSidePisLargeBlk and bSideQisLargeBlk, the condition 1 is defined as follows.

Condition1 = (bSidePisLargeBlk || bSideQisLargeBlk) ? TRUE : FALSE

[0111] Next, if Condition 1 is true, the condition 2 will be further checked. First, the following variables are derived:

- dp0, dp3, dq0, dq3 are first derived as in HEVC
- if (p side is greater than or equal to 32)
 - dp0 = (dp0 + Abs(p5₀ - 2 * p4₀ + p3₀) + 1) >> 1
 - dp3 = (dp3 + Abs(p5₃ - 2 * p4₃ + p3₃) + 1) >> 1
- if (q side is greater than or equal to 32)
 - dq0 = (dq0 + Abs(q5₀ - 2 * q4₀ + q3₀) + 1) >> 1
 - dq3 = (dq3 + Abs(q5₃ - 2 * q4₃ + q3₃) + 1) >> 1

-continued

Condition2 = (d < β) ? TRUE: FALSE
where d = dp0 + dq0 + dp3 + dq3.

[0112] If Condition1 and Condition2 are valid, whether any of the blocks uses sub-blocks is further checked:

```

If (bSidePisLargeBlk)
{
    If (mode block P == SUBBLOCKMODE)
        Sp = 5
    else
        Sp = 7
}
else
    Sp = 3
If (bSideQisLargeBlk)
{
    If (mode block Q == SUBBLOCKMODE)
        Sq = 5
    else
        Sq = 7
}
else
    Sq = 3

```

[0113] Finally, if both the Condition 1 and Condition 2 are valid, the proposed deblocking method will check the condition 3 (the large block strong filter condition), which is defined as follows.

[0114] In the Condition3 StrongFilterCondition, the following variables are derived:

dpq is derived as in HEVC.
 $sp_3 = \text{Abs}(p_3 - p_0)$, derived as in HEVC
 if (p side is greater than or equal to 32)
 if (Sp == 5)
 $sp_3 = (sp_3 + \text{Abs}(p_5 - p_3) + 1) \gg 1$
 else
 $sp_3 = (sp_3 + \text{Abs}(p_7 - p_3) + 1) \gg 1$
 $sq_3 = \text{Abs}(q_0 - q_3)$, derived as in HEVC
 if (q side is greater than or equal to 32)
 if (Sq == 5)
 $sq_3 = (sq_3 + \text{Abs}(q_5 - q_3) + 1) \gg 1$
 else
 $sq_3 = (sq_3 + \text{Abs}(q_7 - q_3) + 1) \gg 1$

[0115] As in HEVC, StrongFilterCondition = (dpq is less than (β >> 2), $sp_3 + sq_3$ is less than (3 * β >> 5), and $\text{Abs}(p_0 - q_0)$ is less than (5 * t_c + 1) >> 1) ? TRUE: FALSE.

2.4.4. Stronger Deblocking Filter for Luma (Designed for Larger Blocks)

[0116] Bilinear filter is used when samples at either one side of a boundary belong to a large block. A sample belonging to a large block is defined as when the width ≥ 32 for a vertical edge, and when height ≥ 32 for a horizontal edge.

[0117] The bilinear filter is listed below.

[0118] Block boundary samples pi for i=0 to Sp-1 and qi for j=0 to Sq-1 (pi and qi are the i-th sample within a row for filtering vertical edge, or the i-th sample within a column for filtering horizontal edge) in HEVC deblocking described above) are then replaced by linear interpolation as follows:

$$p'_i = (f_i * \text{Middle}_{s,t} + (64 - f_i) * P_s + 32) \gg 6, \text{ clipped to } p_i \pm \text{tcPD}_i$$

$$q'_j = (g_j * \text{Middle}_{s,t} + (64 - g_j) * Q_s + 32) \gg 6, \text{ clipped to } q_j \pm \text{tcPD}_j$$

where tcPD_i and tcPD_j term is a position dependent clipping described in Section 2.4.7 and g_j , f_i , $\text{Middle}_{s,t}$, P_s and Q_s are given below.

2.4.5. Deblocking Control for Chroma

[0119] The chroma strong filters are used on both sides of the block boundary. Here, the chroma filter is selected when both sides of the chroma edge are greater than or equal to 8 (chroma position), and the following decision with three conditions are satisfied: the first one is for decision of boundary strength as well as large block. The proposed filter can be applied when the block width or height which orthogonally crosses the block edge is equal to or larger than 8 in chroma sample domain. The second and third one is basically the same as for HEVC luma deblocking decision, which are on/off decision and strong filter decision, respectively.

[0120] In the first decision, boundary strength (bS) is modified for chroma filtering and the conditions are checked sequentially. If a condition is satisfied, then the remaining conditions with lower priorities are skipped.

[0121] Chroma deblocking is performed when bS is equal to 2, or bS is equal to 1 when a large block boundary is detected.

[0122] The second and third condition is basically the same as HEVC luma strong filter decision as follows.

[0123] In the second condition:

[0124] d is then derived as in HEVC luma deblocking.

[0125] The second condition will be TRUE when d is less than β .

[0126] In the third condition StrongFilterCondition is derived as follows:

[0127] dpq is derived as in HEVC.

[0128] $\text{sp}_3 = \text{Abs}(p_3 - p_0)$, derived as in HEVC.

[0129] $\text{sq}_3 = \text{Abs}(q_3 - q_0)$, derived as in HEVC.

[0130] As in HEVC design, StrongFilterCondition = (dpq is less than $(\beta \gg 2)$, $\text{sp}_3 + \text{sq}_3$ is less than $(\beta \gg 3)$, and $\text{Abs}(p_0 - q_0)$ is less than $(5 * t_c + 1) \gg 1$).

2.4.6. Strong Deblocking Filter for Chroma

[0131] The following strong deblocking filter for chroma is defined:

$$\begin{aligned} p'_2 &= (3 * p_3 + 2 * p_2 + p_1 + p_0 + q_0 + 4) \gg 3 \\ p'_1 &= (2 * p_3 + p_2 + 2 * p_1 + p_0 + q_0 + q_1 + 4) \gg 3 \\ p'_0 &= (p_3 + p_2 + p_1 + 2 * p_0 + q_0 + q_1 + q_2 + 4) \gg 3. \end{aligned}$$

[0132] The proposed chroma filter performs deblocking on a 4x4 chroma sample grid.

2.4.7. Position Dependent Clipping

[0133] The position dependent clipping tcPD is applied to the output samples of the luma filtering process involving strong and long filters that are modifying 7, 5 and 3 samples at the boundary. Assuming quantization error distribution, it

is proposed to increase clipping value for samples which are expected to have higher quantization noise, thus expected to have higher deviation of the reconstructed sample value from the true sample value.

[0134] For each P or Q boundary filtered with asymmetrical filter, depending on the result of decision-making process in section 2.4.2, position dependent threshold table is selected from two tables (i.e., Tc7 and Tc3 tabulated below) that are provided to decoder as a side information:

$$\begin{aligned} \text{Tc7} &= \{ 6, 5, 4, 3, 2, 1, 1 \}; \text{Tc3} = \{ 6, 4, 2 \}; \\ \text{tcPD} &= (\text{Sp} = 3) ? \text{Tc3} : \text{Tc7}; \\ \text{tcQD} &= (\text{Sq} = 3) ? \text{Tc3} : \text{Tc7}; \end{aligned}$$

[0135] For the P or Q boundaries being filtered with a short symmetrical filter, position dependent threshold of lower magnitude is applied:

$$\text{Tc3} = \{ 3, 2, 1 \};$$

[0136] Following defining the threshold, filtered p'_i and q'_j sample values are clipped according to tcP and tcQ clipping values:

$$\begin{aligned} p''_i &= \text{Clip3}(p'_i + \text{tcP}_i, p'_i - \text{tcP}_i, p'_i); \\ q''_j &= \text{Clip3}(q'_j + \text{tcQ}_j, q'_j - \text{tcQ}_j, q'_j); \end{aligned}$$

where p'_i and q'_j are filtered sample values, p''_i and q''_j are output sample value after the clipping and tcP_i , tcQ_j are clipping thresholds that are derived from the VVC tc parameter and tcPD and tcQD . The function Clip3 is a clipping function as it is specified in VVC.

2.4.8. Sub-Block Deblocking Adjustment

[0137] To enable parallel friendly deblocking using both long filters and sub-block deblocking the long filters is restricted to modify at most 5 samples on a side that uses sub-block deblocking (AFFINE or ATMVP or DMVR) as shown in the luma control for long filters. Additionally, the sub-block deblocking is adjusted such that that sub-block boundaries on an 8x8 grid that are close to a CU or an implicit TU boundary is restricted to modify at most two samples on each side.

[0138] Following applies to sub-block boundaries that not are aligned with the CU boundary.

```

If (mode block Q == SUBBLOCKMODE && edge != 0) {
  if (!(implicitTU && (edge == (64 / 4))))
    if (edge == 2 || edge == (orthogonalLength - 2) || edge == (56 / 4) ||
        edge == (72 / 4))
      Sp = Sq = 2;
    else
      Sp = Sq = 3;
  else
    Sp = Sq = bSideQisLargeBlk ? 5:3;
}

```

[0139] Where edge equal to 0 corresponds to CU boundary, edge equal to 2 or equal to orthogonalLength-2 corresponds to sub-block boundary 8 samples from a CU boundary etc. Where implicit TU is true if implicit split of TU is used.

2.5. SAO

[0140] The input of SAO is the reconstructed samples after DB. The concept of SAO is to reduce mean sample distortion of a region by first classifying the region samples into multiple categories with a selected classifier, obtaining an offset for each category, and then adding the offset to each sample of the category, where the classifier index and the offsets of the region are coded in the bitstream. In HEVC and VVC, the region (the unit for SAO parameters signaling) is defined to be a CTU.

[0141] Two SAO types that can satisfy the requirements of low complexity are adopted in HEVC. Those two types are edge offset (EO) and band offset (BO), which are discussed in further detail below. An index of an SAO type is coded (which is in the range of [0, 2]). For EO, the sample classification is based on comparison between current samples and neighboring samples according to 1-D directional patterns: horizontal, vertical, 135° diagonal, and 45° diagonal.

[0142] FIG. 11A illustrates an example diagram 1100 showing a 1-D directional pattern for EO sample classification with horizontal (EO class=0). FIG. 11B illustrates an example diagram 1120 showing a 1-D directional pattern for EO sample classification with vertical (EO class=1). FIG. 11C illustrates an example diagram 1140 showing a 1-D directional pattern for EO sample classification with 135° diagonal (EO class=2). FIG. 11D illustrates an example diagram 1160 showing a 1-D directional pattern for EO sample classification with 45° diagonal (EO class=3).

[0143] For a given EO class, each sample inside the CTB is classified into one of five categories. The current sample value, labeled as “c,” is compared with its two neighbors along the selected 1-D pattern. The classification rules for each sample are summarized in Table 1. Categories 1 and 4 are associated with a local valley and a local peak along the selected 1-D pattern, respectively. Categories 2 and 3 are associated with concave and convex corners along the selected 1-D pattern, respectively. If the current sample does not belong to EO categories 1-4, then it is category 0 and SAO is not applied.

TABLE 3

Sample Classification Rules for Edge Offset	
Category	Condition
1	$c < a$ and $c < b$
2	$(c < a \ \&\& \ c = b) \parallel (c = a \ \&\& \ c < b)$
3	$(c > a \ \&\& \ c = b) \parallel (c = a \ \&\& \ c > b)$
4	$c > a \ \&\& \ c > b$
5	None of above

2.6. Geometry Transformation-Based Adaptive Loop Filter in JEM

[0144] The input of DB is the reconstructed samples after DB and SAO. The sample classification and filtering process are based on the reconstructed samples after DB and SAO.

[0145] In the JEM, a geometry transformation-based adaptive loop filter (GALF) with block-based filter adaption is applied. For the luma component, one among 25 filters is selected for each 2×2 block, based on the direction and activity of local gradients.

2.6.1. Filter Shape

[0146] FIG. 12A illustrates an example diagram 1200 showing examples of GALF filter shapes with 5×5 diamond. FIG. 12B illustrates an example diagram 1220 showing examples of GALF filter shapes with 7×7 diamond. FIG. 12C illustrates an example diagram 1240 showing examples of GALF filter shapes with 9×9 diamond.

[0147] In the JEM, up to three diamond filter shapes (as shown in FIGS. 12A-12C) can be selected for the luma component. An index is signalled at the picture level to indicate the filter shape used for the luma component. Each square represents a sample, and Ci (i being 0~6 (left), 0~12 (middle), 0~20 (right)) denotes the coefficient to be applied to the sample. For chroma components in a picture, the 5×5 diamond shape is always used.

2.6.1.1. Block Classification

[0148] Each 2×2 block is categorized into one out of 25 classes. The classification index C is derived based on its directionality D and a quantized value of activity \hat{A} , as follows:

$$C = 5D + \hat{A}. \quad (1)$$

[0149] To calculate D and \hat{A} , gradients of the horizontal, vertical and two diagonal direction are first calculated using 1-D Laplacian:

$$g_v = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} V_{k,l}, V_{k,l} = |2R(k, l) - R(k, l-1) - R(k, l+1)|, \quad (2)$$

$$g_h = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} H_{k,l}, H_{k,l} = |2R(k, l) - R(k-1, l) - R(k+1, l)|, \quad (3)$$

$$g_{d1} = \sum_{k=i-2}^{i+3} \sum_{l=j-3}^{j+3} D1_{k,l}, D1_{k,l} = \quad (4)$$

$$|2R(k, l) - R(k-1, l-1) - R(k+1, l+1)|$$

$$g_{d2} = \sum_{k=i-2}^{i+3} \sum_{l=j-2}^{j+3} D2_{k,l}, D2_{k,l} = \quad (5)$$

$$|2R(k, l) - R(k-1, l+1) - R(k+1, l-1)|$$

[0150] Indices i and j refer to the coordinates of the upper left sample in the 2×2 block and R(i,j) indicates a reconstructed sample at coordinate (i,j).

[0151] Then D maximum and minimum values of the gradients of horizontal and vertical directions are set as:

$$g_{h,v}^{max} = \max(g_h, g_v), g_{h,v}^{min} = \min(g_h, g_v), \quad (6)$$

and the maximum and minimum values of the gradient of two diagonal directions are set as:

$$g_{d0,d1}^{max} = \max(g_{d0}, g_{d1}), g_{d0,d1}^{min} = \min(g_{d0}, g_{d1}), \quad (7)$$

[0152] To derive the value of the directionality D, these values are compared against each other and with two thresholds t_1 and t_2 :

[0153] Step 1. If both $g_{h,v}^{max} \leq t_1 \cdot g_{h,v}^{min}$ and $g_{d0,d1}^{max} \leq t_1 \cdot g_{d0,d1}^{min}$ are true, D is set to 0.

[0154] Step 2. If $g_{h,v}^{max}/g_{h,v}^{min} > g_{d0,d1}^{max}/g_{d0,d1}^{min}$, continue from Step 3; otherwise continue from Step 4.

[0155] Step 3. If $g_{h,v}^{max} > t_2 \cdot g_{h,v}^{min}$, D is set to 2; otherwise D is set to 1.

[0156] Step 4. If $g_{d0,d1}^{max} > t_2 \cdot g_{d0,d1}^{min}$, D is set to 4; otherwise D is set to 3.

[0157] The activity value A is calculated as:

$$A = \sum_{k=-2}^{i+3} \sum_{l=-2}^{j+3} (V_{k,l} + H_{k,l}). \quad (8)$$

[0158] A is further quantized to the range of 0 to 4, inclusively, and the quantized value is denoted as \hat{A} .

[0159] For both chroma components in a picture, no classification method is applied, i.e. a single set of ALF coefficients is applied for each chroma component.

2.6.1.2. Geometric Transformations of Filter Coefficients

[0160] FIG. 13A illustrates an example diagram 1300 showing relative coordinator for the 5x5 diamond filter support (diagonal). FIG. 13B illustrates an example diagram 1320 showing relative coordinator for the 5x5 diamond filter support (vertical flip). FIG. 13C illustrates an example diagram 1340 showing relative coordinator for the 5x5 diamond filter support (rotation).

[0161] Before filtering each 2x2 block, geometric transformations such as rotation or diagonal and vertical flipping are applied to the filter coefficients $f(k, l)$, which is associated with the coordinate (k, l) , depending on gradient values calculated for that block. This is equivalent to applying these transformations to the samples in the filter support region. The idea is to make different blocks to which ALF is applied more similar by aligning their directionality.

[0162] Three geometric transformations, including diagonal, vertical flip and rotation are introduced:

$$\text{Diagonal: } f_D(k, l) = f(l, k), \quad (9)$$

$$\text{Vertical flip: } f_V(k, l) = f(k, K - l - 1),$$

$$\text{Rotation: } f_R(k, l) = f(K - l - 1, k).$$

where K is the size of the filter and $0 \leq k, l \leq K-1$ are coefficients coordinates, such that location (0,0) is at the upper left corner and location (K-1, K-1) is at the lower right corner. The transformations are applied to the filter coefficients $f(k, l)$ depending on gradient values calculated for that block. The relationship between the transformation and the four gradients of the four directions are summarized in Table 4. FIGS. 12A-12C show the transformed coefficients for each position based on the 5x5 diamond.

TABLE 4

Mapping of the gradient calculated for one block and the transformations	
Gradient values	Transformation
$g_{d2} < g_{d1}$ and $g_h < g_v$	No transformation
$g_{d2} < g_{d1}$ and $g_v < g_h$	Diagonal
$g_{d1} < g_{d2}$ and $g_h < g_v$	Vertical flip
$g_{d1} < g_{d2}$ and $g_v < g_h$	Rotation

2.6.1.3. Filter Parameters Signalling

[0163] In the JEM, GALF filter parameters are signalled for the first CTU, i.e., after the slice header and before the SAO parameters of the first CTU. Up to 25 sets of luma filter coefficients could be signalled. To reduce bits overhead, filter coefficients of different classification can be merged. Also, the GALF coefficients of reference pictures are stored and allowed to be reused as GALF coefficients of a current picture. The current picture may choose to use GALF coefficients stored for the reference pictures and bypass the GALF coefficients signalling. In this case, only an index to one of the reference pictures is signalled, and the stored GALF coefficients of the indicated reference picture are inherited for the current picture.

[0164] To support GALF temporal prediction, a candidate list of GALF filter sets is maintained. At the beginning of decoding a new sequence, the candidate list is empty. After decoding one picture, the corresponding set of filters may be added to the candidate list. Once the size of the candidate list reaches the maximum allowed value (i.e., 6 in current JEM), a new set of filters overwrites the oldest set in decoding order, and that is, first-in-first-out (FIFO) rule is applied to update the candidate list. To avoid duplications, a set could only be added to the list when the corresponding picture doesn't use GALF temporal prediction. To support temporal scalability, there are multiple candidate lists of filter sets, and each candidate list is associated with a temporal layer. More specifically, each array assigned by temporal layer index (TempIdx) may compose filter sets of previously decoded pictures with equal to lower TempIdx. For example, the k-th array is assigned to be associated with TempIdx equal to k, and it only contains filter sets from pictures with TempIdx smaller than or equal to k. After coding a certain picture, the filter sets associated with the picture will be used to update those arrays associated with equal or higher TempIdx.

[0165] Temporal prediction of GALF coefficients is used for inter coded frames to minimize signalling overhead. For intra frames, temporal prediction is not available, and a set of 16 fixed filters is assigned to each class. To indicate the usage of the fixed filter, a flag for each class is signalled and if required, the index of the chosen fixed filter. Even when the fixed filter is selected for a given class, the coefficients of the adaptive filter $f(k, l)$ can still be sent for this class in which case the coefficients of the filter which will be applied to the reconstructed image are sum of both sets of coefficients.

[0166] The filtering process of luma component can controlled at CU level. A flag is signalled to indicate whether GALF is applied to the luma component of a CU. For chroma component, whether GALF is applied or not is indicated at picture level only.

2.6.1.4. Filtering Process

[0167] At decoder side, when GALF is enabled for a block, each sample $R(i, j)$ within the block is filtered, resulting in sample value $R'(i, j)$ as shown below, where L denotes filter length, $f_{m,n}$ represents filter coefficient, and $f(k, l)$ denotes the decoded filter coefficients.

$$R'(i, j) = \sum_{k=-L/2}^{L/2} \sum_{l=-L/2}^{L/2} f(k, l) \times R(i+k, j+l) \quad (10)$$

[0168] FIG. 14 illustrates an example diagram 1400 showing examples of relative coordinates for the 5x5 diamond filter support. FIG. 14 shows an example of relative coordinates used for 5x5 diamond filter support supposing the current sample's coordinate (i, j) to be $(0, 0)$. Samples in different coordinates filled with the same color are multiplied with the same filter coefficients.

2.7. Geometry Transformation-Based Adaptive Loop Filter (GALF) in VVC

2.7.1. GALF in VTM-4

[0169] In VTM4.0, the filtering process of the Adaptive Loop Filter, is performed as follows:

$$O(x, y) = \sum_{(i,j)} w(i, j) \cdot I(x+i, y+j), \quad (11)$$

where samples $I(x+i, y+j)$ are input samples, $O(x, y)$ is the filtered output sample (i.e. filter result), and $w(i, j)$ denotes the filter coefficients. In practice, in VTM4.0 it is implemented using integer arithmetic for fixed point precision computations:

$$O(x, y) = \left(\sum_{i=-L/2}^{L/2} \sum_{j=-L/2}^{L/2} w(i, j) \cdot I(x+i, y+j) + 64 \right) \gg 7, \quad (12)$$

where L denotes the filter length, and where $w(i, j)$ are the filter coefficients in fixed point precision.

[0170] The current design of GALF in VVC has the following major changes compared to that in JEM:

[0171] 1) The adaptive filter shape is removed. Only 7x7 filter shape is allowed for luma component and 5x5 filter shape is allowed for chroma component.

[0172] 2) Signaling of ALF parameters is removed from slice/picture level to CTU level.

[0173] 3) Calculation of class index is performed in 4x4 level instead of 2x2. In addition, sub-sampled Laplacian calculation method for ALF classification is utilized. More specifically, there is no need to calculate the horizontal/vertical/45 degree diagonal/135 degree gradients for each sample within one block. Instead, 1:2 subsampling is utilized.

2.8. Non-Linear ALF in Current VVC

2.8.1. Filtering Reformulation

[0174] Equation (11) can be reformulated, without coding efficiency impact, in the following expression:

$$O(x, y) = I(x, y) + \sum_{(i,j) \neq (0,0)} w(i, j) \cdot (I(x+i, y+j) - I(x, y)), \quad (13)$$

where $w(i, j)$ are the same filter coefficients as in equation (11) [excepted $w(0, 0)$ which is equal to 1 in equation (13) while it is equal to $1 - \sum_{(i,j) \neq (0,0)} w(i, j)$ in equation (11)].

[0175] Using this above filter formula of (13), VVC introduces the non-linearity to make ALF more efficient by using a simple clipping function to reduce the impact of neighbor sample values $(I(x+i, y+j))$ when they are too different with the current sample value $(I(x, y))$ being filtered.

[0176] More specifically, the ALF filter is modified as follows:

$$O'(x, y) = \quad (14)$$

$$I(x, y) + \sum_{(i,j) \neq (0,0)} w(i, j) \cdot K(I(x+i, y+j) - I(x, y), k(i, j)),$$

where $K(d, b) = \min(b, \max(-b, d))$ is the clipping function, and $k(i, j)$ are clipping parameters, which depends on the (i, j) filter coefficient. The encoder performs the optimization to find the best $k(i, j)$. In some implementation, the clipping parameters $k(i, j)$ are specified for each ALF filter, one clipping value is signaled per filter coefficient. It means that up to 12 clipping values can be signalled in the bitstream per Luma filter and up to 6 clipping values for the Chroma filter.

[0177] In order to limit the signaling cost and the encoder complexity, only 4 fixed values which are the same for INTER and INTRA slices are used.

[0178] Because the variance of the local differences is often higher for Luma than for Chroma, two different sets for the Luma and Chroma filters are applied. The maximum sample value (here 1024 for 10 bits bit-depth) in each set is also introduced, so that clipping can be disabled if it is not necessary.

[0179] The sets of clipping values are provided in the Table 5. The 4 values have been selected by roughly equally splitting, in the logarithmic domain, the full range of the sample values (coded on 10 bits) for Luma, and the range from 4 to 1024 for Chroma.

[0180] More precisely, the Luma table of clipping values have been obtained by the following formula:

$$AlfClip_L = \left\{ \text{round} \left(\left(\frac{M}{A} \right)^{\frac{1}{N-1}} \right)^{N-n+1} \right\} \text{ for } n \in 1..N, \quad (15)$$

with $M = 2^{10}$ and $N = 4$.

[0181] Similarly, the Chroma tables of clipping values is obtained according to the following formula:

$$AlfClip_C = \left\{ \text{round} \left(A \cdot \left(\frac{M}{A} \right)^{\frac{1}{N-1}} \right)^{N-n} \right\} \text{ for } n \in 1..N, \quad (16)$$

with $M = 2^{10}$, $N = 4$ and $A = 4$.

TABLE 5

Authorized clipping values	
INTRA/INTER tile group	
LUMA	{ 1024, 181, 32, 6 }
CHROMA	{ 1024, 161, 25, 4 }

[0182] The selected clipping values are coded in the “alf_data” syntax element by using a Golomb encoding scheme corresponding to the index of the clipping value in the above Table 5. This encoding scheme is the same as the encoding scheme for the filter index.

2.9. Convolutional Neural Network-Based Loop Filters for Video Coding

2.9.1. Convolutional Neural Networks

[0183] In deep learning, a convolutional neural network (CNN, or ConvNet) is a class of deep neural networks, most commonly applied to analyzing visual imagery. They have very successful applications in image and video recognition/processing, recommender systems, image classification, medical image analysis, natural language processing.

[0184] CNNs are regularized versions of multilayer perceptrons. Multilayer perceptrons usually mean fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer. The “fully-connectedness” of these networks makes them prone to overfitting data. Typical ways of regularization include adding some form of magnitude measurement of weights to the loss function. CNNs take a different approach towards regularization: they take advantage of the hierarchical pattern in data and assemble more complex patterns using smaller and simpler patterns. Therefore, on the scale of connectedness and complexity, CNNs are on the lower extreme.

[0185] CNNs use relatively little pre-processing compared to other image classification/processing algorithms. This means that the network learns the filters that in traditional algorithms were hand-engineered. This independence from prior knowledge and human effort in feature design is a major advantage.

2.9.2. Deep Learning for Image/Video Coding

[0186] Deep learning-based image/video compression typically has two implications: end-to-end compression purely based on neural networks and traditional frameworks enhanced by neural networks. The first type usually takes an auto-encoder like structure, either achieved by convolutional neural networks or recurrent neural networks. While purely relying on neural networks for image/video compression can avoid any manual optimizations or hand-crafted designs, compression efficiency may be not satisfactory. Therefore, works distributed in the second type take neural networks as an auxiliary, and enhance traditional compression frameworks by replacing or enhancing some modules. In this way, they can inherit the merits of the highly optimized traditional frameworks. For example, a fully connected network for the intra prediction is proposed. In addition to intra prediction, deep learning is also exploited to enhance other modules. For example, the in-loop filters of HEVC with a convolu-

tional neural network is replaced and promising results are achieved. Neural networks are applied to improve the arithmetic coding engine.

2.9.3. Convolutional Neural Network Based In-Loop Filtering

[0187] In lossy image/video compression, the reconstructed frame is an approximation of the original frame, since the quantization process is not invertible and thus incurs distortion to the reconstructed frame. To alleviate such distortion, a convolutional neural network could be trained to learn the mapping from the distorted frame to the original frame. In practice, training must be performed prior to deploying the CNN-based in-loop filtering.

2.9.3.1. Training

[0188] The purpose of the training processing is to find the optimal value of parameters including weights and bias.

[0189] First, a codec (e.g. HM, JEM, VTM, etc.) is used to compress the training dataset to generate the distorted reconstruction frames.

[0190] Then the reconstructed frames are fed into the CNN and the cost is calculated using the output of CNN and the groundtruth frames (original frames). Commonly used cost functions include SAD (Sum of Absolution Difference) and MSE (Mean Square Error). Next, the gradient of the cost with respect to each parameter is derived through the back propagation algorithm. With the gradients, the values of the parameters can be updated. The above process repeats until the convergence criteria is met. After completing the training, the derived optimal parameters are saved for use in the inference stage.

2.9.3.2. Convolution Process

[0191] During convolution, the filter is moved across the image from left to right, top to bottom, with a one-pixel column change on the horizontal movements, then a one-pixel row change on the vertical movements. The amount of movement between applications of the filter to the input image is referred to as the stride, and it is almost always symmetrical in height and width dimensions. The default stride or strides in two dimensions is (1,1) for the height and the width movement.

[0192] FIG. 15A illustrates an example diagram 1500 showing Architecture of the proposed CNN filter. FIG. 15B illustrates an example diagram 1550 showing a construction of ResBlock (residual block) in the CNN filter. In most of deep convolutional neural networks, residual blocks are utilized as the basic module and stacked several times to construct the final network where in one example, the residual block is obtained by combining a convolutional layer, a ReLU/PReLU activation function and a convolutional layer as shown in FIG. 15B.

2.9.3.3. Inference

[0193] During the inference stage, the distorted reconstruction frames are fed into CNN and processed by the CNN model whose parameters are already determined in the training stage. The input samples to the CNN can be reconstructed samples before or after DB, or reconstructed samples before or after SAO, or reconstructed samples before or after ALF.

3. Problems

[0194] The current NN filter has the following problems:

[0195] 1. The prior art design of NN filter is only applied after the reconstruction of all blocks before in-loop filtering processes within a slice. Therefore, the impact of reduced distortion due to NN filter is not taken into consideration during the rate-distortion optimization (RDO) process, such as intra mode selection, partitioning selection, intra mode selection, inter mode selection, transform core selection, etc. The coding performance is sub-optimal considering:

[0196] a. The best mode (e.g., coding method/partitioning sizes) of current block selected in the RDO process could be wrong since the distortion is calculated without NN filter being applied.

[0197] b. The reconstruction and associated coded information of current block has big impact on coding of the subsequent blocks (e.g., due to intra prediction, or motion prediction). If the current block doesn't select the best mode, then the coding performance of sub-sequence block will also be sub-optimal.

4. Description

[0198] The detailed embodiments below should be considered as examples to explain general concepts. These embodiments should not be interpreted in a narrow way. Furthermore, these embodiments can be combined in any manner.

[0199] To solve the above problem, it is proposed to take the NN filter into consideration based on video content during the rate distortion optimization (RDO) process. The present disclosure elaborates how to extend RDO purview with NN filter models, how to utilize NN filter models to select mode (e.g. intra mode, partitioning mode, inter mode or transform core), how to control the usage of NN filter models.

[0200] In the disclosure, a NN filter can be any kind of NN filter, such as a convolutional neural network (CNN) filter; alternatively, it could also be applied to non-NN based filters. In the following discussion, a NN filter may also be referred to as a CNN filter.

[0201] In the following discussion, a video unit may be a sequence, a picture, a slice, a tile, a brick, a subpicture, a CTU/CTB, a CTU/CTB row, one or multiple CUs/CBs, one or multiple CTUs/CTBs, one or multiple VPDU (Virtual Pipeline Data Unit), a sub-region within a picture/slice/tile/brick. A father video unit represents a unit larger than the video unit. Typically, a father unit will contain several video units. E.g., when the video unit is CTU, the father unit could be slice, CTU row, multiple CTUs, etc.

[0202] The width and height of a video unit are denoted as W and H, respectively.

On Utilization of the NN Filter Models to Select the Modes in the RDO Process

[0203] 1. Whether to and/or how to utilize the NN filter models (or calculate the rate distortion cost) in RDO process may be dependent on the distortion D_{ORG} without NN filter model and/or the distortion D'_{NNLF} with n^{th} NN filter model (the model index is $n-1$, where $n \geq 1$). The RDO criterion is noted as $J = D + \lambda * R$. "A distortion D'_{NNLF} with the n^{th} NN filter" may mean

that the reconstruction samples are filtered by the n^{th} NN filter and the filtered reconstruction samples will be compared with the original samples to derive the distortion.

[0204] a. In one example, D in RDO criterion is the distortion with the n^{th} NN filter model, $D = D'_{NNLF}$.

[0205] i. In one example, n is 1, 2, 3.

[0206] b. In one example, D in RDO criterion is the minimal value of the distortion without NN filter model and distortion with the n^{th} NN filter model. $D = \min(D'_{NNLF}, D_{ORG})$.

[0207] i. In one example, n is 1, 2, 3.

[0208] c. In one example, D in RDO criterion is the distortion with the best one of NN filter models.

[0209] i. In one example, the best NN filter model is selected by distortion.

[0210] ii. In one example, the best NN filter model is default one.

[0211] d. In one example, D in RDO criterion is the minimal value of distortion without NN filter model and the distortion with the best one of NN filter models.

[0212] i. In one example, the best NN filter model is selected by distortion.

[0213] ii. In one example, the best NN filter model is default one.

[0214] e. In one example, D in RDO criterion is a scaled version of the distortion without NN filter model, i.e. $D = f * D_{ORG}$.

[0215] i. In one example, f is 1.0, 0.9, or 1.1.

[0216] f. In one example, D in RDO criterion is derived according to the distortion without NN filter model and the distortion with the n^{th} NN filter model.

[0217] i. In one example, $D = f_0 * D_{ORG} + f_1 * D'_{NNLF}$.

[0218] 1) In one example, $f_0 = 0.0$, $f_1 = 1.0$;

[0219] 2) In one example, $f_0 = 1.0$, $f_1 = 0.0$.

[0220] ii. In one example, $D = \min(f_0 * D_{ORG}, f_1 * D'_{NNLF})$.

[0221] 1) In one example, $f_0 = 1.0$, $f_1 = 1.0$.

[0222] g. In one example, D in RDO criterion may be the combination of multiple calculating methods in the above embodiments.

[0223] i. In one example, the combination of calculating methods may be dependent on the coding statistics of the video unit (e.g., prediction modes, qp, temporal layer, slice type, etc.).

[0224] 1) In one example, the combination of calculating methods may be dependent on the type of candidate modes.

[0225] 2) In one example, when all or one of T candidate modes are NOT partitioning modes, D in RDO criterion is combination of the distortion with the p^{th} NN filter model and/or distortion without NN filter model.

a. In one example, when the p^{th} NN filter model is NOT applied for one or all of T candidate modes, D in RDO criterion is the distortion without NN filter model ($D = D_{ORG}$).

b. In one example, when the p^{th} NN filter model is applied for all or one of T candidate modes, D in RDO criterion is the distortion with the p^{th} NN filter model ($D = D'_{NNLF}$).

[0226] 3) In one example, when all or one of T candidate modes are partitioning modes, D in

RDO criterion is combination of the distortion with the q^{th} NN filter model and/or the distortion with the p^{th} NN filter model and/or distortion without NN filter model.

- [0227] ii. In one example, the combination may be dependent on the usage of the q^{th} NN filter model and/or the usage of the p^{th} NN filter model.
- [0228] 1) In one example, when the q^{th} NN filter model is applied for all or one of T candidate modes, D in RDO criterion is the minimal value of the distortion without NN filter model and distortion with the q^{th} NN filter model ($D = \min(D_{NNLF}^q, D_{ORG})$).
- [0229] 2) In one example, when the q^{th} NN filter model is NOT applied for one or all of T candidate modes, D in RDO criterion is combination of the distortion with the p^{th} NN filter model and distortion without NN filter model.
- a. In one example, when the p^{th} NN filter model is applied for all of T candidate modes, D in RDO criterion is the distortion with the p^{th} NN filter model ($D = D_{NNLF}^p$).
- b. In one example, when the p^{th} NN filter model is NOT applied for one or all of T candidate modes, D in RDO criterion is the distortion without NN filter model ($D = D_{ORG}$).
- [0230] iii. In one example, the methods of calculating the D in RDO criterion may be applied according to the certain or adaptive order.
- [0231] 1) In one example, the order of applying the methods of calculating the D in RDO criterion may be dependent on the coding modes/statistics of the video unit (e.g., prediction modes, qp, temporal layer, slice type, etc.).
- [0232] 2) In one example, $D = \min(D_{NNLF}^q, D_{ORG})$ may have a greater priority than $D = D_{NNLF}^p$.
- [0233] 3) In one example, $D = D_{NNLF}^p$ may have a greater priority than $D = D_{ORG}$.
- [0234] iv. In the above embodiments, the p^{th} NN filter model may be the deblocking filter or SAO or ALF.
- [0235] v. In the above embodiments, the q^{th} NN filter model may be the CNN filter.
- [0236] vi. In the above embodiments, the q^{th} NN filter model may be same with the p^{th} NN filter model.
- [0237] 1) In one example, p may be equal to q.
- [0238] vii. In the above embodiments, T candidate modes may be the comparable modes with the RDO criterion noted as $J = D + \lambda * R$.
- [0239] 1) In one example, the number T of candidate modes may be equal to 1 or 2 or 3 or 4.
- [0240] h. In the above embodiments, the parameters f or f_0 or f_1 may be set according to the temporal layers and/or slice types.
- [0241] i. In the above embodiments, the parameters f or f_0 or f_1 may be set according to the QP (quantization parameter).
- [0242] j. In the above embodiments, the parameters f or f_0 or f_1 may be set according to the configuration (e.g., all intra, random access, low-delay B, low-delay P, etc.).
- [0243] k. In one example, the same rule of usage of NN filtering should be applied to calculate RD costs for all candidate modes and/or partitioning methods for a block.
- [0244] 2. Whether to and/or how to utilize the NN filter models (or calculate the rate distortion cost) in RDO process may be dependent on the distortion D_{ORG} without NN filter model and/or the combination of distortions with multiple NN filter models. The RDO criterion is noted as $J = D + \lambda * R$.
- [0245] a. In one example, D in RDO criterion is the minimal value of distortions with m NN filter models.
- [0246] i. In one example, m is the available number of constructed NN filter models.
- [0247] ii. In one example, m is dependent on the type of video unit.
- [0248] iii. In one example, m is dependent on the signaled parameters of video unit.
- [0249] iv. In one example, m is the default value.
- [0250] v. In one example, m is 0, 1, 2, 3, 4, etc.
- [0251] b. In one example, D in RDO criterion is the minimal value of the distortion without NN filter model and distortions with m NN filter models.
- [0252] i. In one example, m is the available number of constructed NN filter models.
- [0253] ii. In one example, m is dependent on the type of video unit.
- [0254] iii. In one example, m is dependent on the signaled parameters of video unit.
- [0255] iv. In one example, m is the default value.
- [0256] v. In one example, m is 0, 1, 2, 3, 4, etc.
- [0257] c. In one example, D in RDO criterion is combination of the distortion without NN filter model and distortions with m NN filter models.
- [0258] i. In one example, $D = f_0 * D_{ORG} + \sum f_n * D_{NNLF}^n$.
- [0259] d. In the above embodiments, the parameters m or f_0 or f_n may be set according to the temporal layers and/or slice types.
- [0260] e. In the above embodiments, the parameters m or f_0 or f_n may be set according to the QP (quantization parameter).
- [0261] f. In one example, the same rule of usage of NN filtering should be applied to calculate RD costs for all candidate modes and/or partitioning methods for a block.
- [0262] 3. The distortion D_{ORG} without NN filter model may be dependent on the other filters LF^i which are different with the NN filtering models (e.g., Deblocking, ALF). The distortion D_{LF}^i with LF^i may mean that the reconstruction samples are filtered by the LF^i filter and the filtered reconstruction samples will be compared with the original samples to derive the distortion.
- [0263] a. In one example, the LF^i may be applied on the reconstruction samples before NN filtering models.
- [0264] b. In one example, the NN filtering models may be applied after LF^i .
- [0265] c. In one example, D_{ORG} is a scaled version of the distortion D_{LF}^i with filter LF , i.e., $D_{ORG} = f * D_{LF}^i$.
- [0266] d. In one example, D_{ORG} is the combination of the distortions with multiple of other filters, i.e., $D_{ORG} = \sum f_i * D_{LF}^i$.

On Implementation of the NN Filter Models in the RDO Process

- [0267] 4. The input of the NN filter models in RDO process may include the signal from the current video block and/or the neighboring blocks.
- [0268] a. In one example, the prediction signal and/or partitioning information and/or reconstruction signal from the current block may be involved in the NN filter process.
- [0269] b. In one example, other information from one/multiple reference frames may be involved in the NN filter process of current block.
- [0270] i. In one example, the collocated block from the first frame in list-0 and/or the collocated block from the first frame in list-1 may be involved in the NN filter process.
- [0271] ii. In one example, one/multiple motion compensated reference blocks may be involved in the NN filter process.
- [0272] c. In one example, the prediction signal and/or partitioning information and/or pixels from at least one neighboring block may be involved in the NN filter process.
- [0273] i. In one example, the neighboring blocks may be located at the left or above or left-above side of the current block.
- [0274] ii. In one example, the samples of neighboring blocks may be reconstructed.
- [0275] iii. In one example, The neighboring blocks may be located at the right or below side of the current block.
- [0276] iv. In one example, the samples of neighboring blocks may not be reconstructed.
- [0277] v. In one example, the samples of neighboring blocks may be original signal.
- [0278] vi. In one example, the samples of neighboring blocks may be derived from the current block.
- [0279] d. In one example, these above embodiments could be combined in any manner.

On Usage of the NN Filter Models in the RDO Process

- [0280] 5. Whether to and/or how to utilize the NN filter models in RDO process may be dependent on the coding statistics of the video unit (e.g., prediction modes, qp, temporal layer, slice type, etc.).
- [0281] a. Whether to and/or how to utilize the NN filter models in RDO process may be dependent on the dimension of the video unit.
- [0282] i. In one example, when $W \leq T1$ and/or $H \leq T2$ (e.g., $T1=T2=64$), the NN filter models may be utilized in the RDO process.
- [0283] ii. In one example, when $W \geq T1$ and/or $H \geq T2$ (e.g., $T1=T2=16$), the NN filter models may be utilized in the RDO process.
- [0284] iii. In one example, when $W \cdot H \geq T1$ (e.g., $T1=64$), the NN filter models may be utilized in the RDO process.
- [0285] iv. In one example, when $W \cdot H \leq T1$ (e.g., $T1=4096$), the NN filter models may be utilized in the RDO process.

- [0286] b. Whether to and/or how to utilize the NN filter models in RDO process may be dependent on the color components.

- [0287] i. In one example, the NN filter models is only applied in the component X.

- [0288] 1) In one example, X is Luma (Y) or Chroma (Cb or Cr).

- [0289] ii. In one example, the NN filter models is applied for all color components.

- [0290] c. Whether to and/or how to utilize the NN filter models in RDO process may be dependent on the rate distortion cost without NN filter model. J_A means the cost of mode A without NN filter model and J_B means the cost of mode B without NN filter model.

- [0291] i. In one example, NN filter models may be not applied when the cost J_A is greater than and/or equal than multiple of cost J_B ($J_A > f_0 \cdot J_B$ or $J_A \geq f_0 \cdot J_B$).

- [0292] 1) In one example, f_0 is 1.0, 1.001, 1.005, 1.05, 1.01.

- [0293] ii. In one example, NN filter models may be not applied when the cost J_B is greater than and/or equal than multiple of cost J_A ($J_B > f_1 \cdot J_A$ or $J_B \geq f_1 \cdot J_A$).

- [0294] 1) In one example, f_1 is 1.0, 1.001, 1.005, 1.05, 1.01.

- [0295] iii. In one example, $J_A > f_0 \cdot J_B$ || $J_B > f_1 \cdot J_A$.

- [0296] iv. In one example, NN filter models may be not applied when the ratio between cost J_A and cost J_B is greater than and/or equal than a threshold ($J_A/J_B > f_2$ or $J_A/J_B \geq f_2$).

- [0297] 1) In one example, f_2 is 1.0, 1.001, 1.005, 1.05, 1.01.

- [0298] v. In one example, NN filter models may be not applied when the ratio between cost J_A and cost J_B is smaller than and/or equal than a threshold ($J_A/J_B < f_3$ or $J_A/J_B \leq f_3$).

- [0299] 1) In one example, f_3 is 1.0, 1.001, 1.005, 1.05, 1.01.

- [0300] vi. In one example, $J_A/J_B > f_2$ || $J_A/J_B < f_3$.

- [0301] vii. In one example, NN filter models may be NOT or be applied when the cost J_A is equal or/and smaller or/and greater than a threshold Th_A and/or the cost J_B is equal or/and smaller or/and greater than a threshold Th_B .

- [0302] 1) In one example, Th_A or The is equal to MAX_DOUBLE which is a default value.

- [0303] 2) In one example, Th_A or The is equal to $1.7e+308$.

- [0304] viii. In one example, the NN filter for mode A and mode B may be applied according to the certain or adaptive order.

- [0305] 1) In one example, the order of applying the NN filter for mode A and mode B may be dependent on the coding modes/statistics of the video unit (e.g., prediction modes, qp, temporal layer, slice type, etc.).

- [0306] 2) In one example, the order of applying the NN filter for mode A and mode B may be dependent on J_A and J_B .

- a. In one example, when J_A is greater than and/or equal than J_B , applying the NN filter for

- mode A may have a greater priority than applying the NN filter for mode B.
- b. In one example, when J_B is greater than and/or equal than J_A , applying the NN filter for mode A may have a greater priority than applying the NN filter for mode B.
- [0307] ix. In one example, J_A may be equal to $D + \lambda R$.
- [0308] 1) In one example, D may be equal to D_{ORG} .
- [0309] x. In one example, J_B may be equal to $D + \lambda R$.
- [0310] 1) In one example, D may be equal to D_{ORG} .
- [0311] xi. In one example, these above embodiments could be combined in any manner.
- [0312] xii. In the above embodiments, the parameters Th_A , Th_B , f_0 , f_1 , f_2 , or f_3 may be set according to the temporal layers.
- [0313] xiii. In the above embodiments, the parameters Th_A , Th_B , f_0 , f_1 , f_2 , or f_3 may be set according to the QP (quantization parameter).
- [0314] xiv. In the above embodiments, the parameters Th_A , Th_B , f_0 , f_1 , f_2 , or f_3 may be set according to the slice type.
- [0315] xv. In the above embodiments, the parameters Th_A , Th_B , f_0 , f_1 , f_2 , or f_3 may be set according to the configuration (e.g., all intra, random access, low-delay B, low-delay P, etc.).
- [0316] d. Whether to and/or how to utilize the NN filter models in RDO process may be dependent on the rate distortion cost without NN filter model and the rate distortion cost with NN filter model. J_A means the cost of mode A without NN filter model and J'_B means the cost of mode B with NN filter model.
- [0317] i. In one example, whether to apply NN filter models for mode A may be dependent on the cost J_A and J'_B .
- [0318] ii. In one example, J_A may be equal to $D + \lambda R$.
- [0319] 1) In one example, D may be equal to D_{ORG} .
- [0320] iii. In one example, J'_B may be equal to $D + \lambda R$.
- [0321] 1) In one example, D may be equal to $\min(D_{NNLF}, D_{ORG})$.
- [0322] 2) In one example, D may be equal to D_{NNLF} .
- [0323] iv. In one example, NN filter models may be not applied for mode A when the cost J_A is greater than and/or equal than multiple of cost J'_B ($J_A > f_0 J'_B$ or $J_A \geq f_0 J'_B$).
- [0324] v. In one example, NN filter models may be not applied for mode A when the cost J'_B is greater than and/or equal than multiple of cost J_A ($J'_B > f_1 J_A$ or $J'_B \geq f_1 J_A$).
- [0325] vi. In one example, $J_A > f_0 + J'_B$ or $J'_B > f_1 J_A$.
- [0326] vii. In one example, NN filter models may be not applied when the ratio between cost J_A and cost J'_B is greater than and/or equal than a threshold ($J_A/J'_B > f_2$ or $J_A/J'_B \geq f_2$).
- [0327] viii. In one example, NN filter models may be not applied when the ratio between cost J_A and cost J'_B is smaller than and/or equal than a threshold ($J_A/J'_B < f_3$ or $J_A/J'_B \leq f_3$).
- [0328] ix. In one example, $J_A/J'_B > f_2$ or $J_A/J'_B < f_3$.
- [0329] x. In one example, these above embodiments could be combined in any manner.
- [0330] xi. In the above embodiments, the parameters f_0 , f_1 , f_2 , or f_3 may be set according to the temporal layers.
- [0331] xii. In the above embodiments, the parameters f_0 , f_1 , f_2 , or f_3 may be set according to the QP (quantization parameter).
- [0332] xiii. In the above embodiments, the parameters f_0 , f_1 , f_2 , or f_3 may be set according to the slice type.
- [0333] xiv. In the above embodiments, the parameters f_0 , f_1 , f_2 , or f_3 may be set according to the configuration (e.g., all intra, random access, low-delay B, low-delay P, etc.).
- [0334] e. In one example, NN filtering may only be applied to partial samples of a block in the RDO process.
- [0335] i. For example, NN filtering may only be applied to the center $W1 \times H1$ subblock in a $W \times H$ block.
- [0336] 1) For example, $W1 = W/2$, $H1 = H/2$.
- [0337] 2) For example, $W1 = 3 \times W/4$, $H1 = 3 \times H/4$.
- [0338] f. Whether to and/or how to utilize the NN filter models in RDO process may be dependent on the temporal layers.
- [0339] i. In one example, NN filter models may be applied to temporal layers with ID greater than K.
- [0340] 1) In one example, K is equal to 0, 1, 2, 3, 4, 5, 6.
- [0341] ii. In one example, NN filter models may be applied to temporal layers with ID smaller than K.
- [0342] 1) In one example, K is equal to 0, 1, 2, 3, 4, 5, 6.
- [0343] g. Whether to and/or how to utilize the NN filter models in RDO process may be dependent on the coding statistics of sub coding units.
- [0344] i. In one example, NN filter models may be NOT or be applied for the current block when NN filter models are applied to part or all of sub coding units.
- [0345] ii. In one example, NN filter models may be NOT or be applied for the current block when rate-distortion costs and/or the distortions of part or all of sub coding units are available.

5. Embodiment

5.1 Embodiment #1

[0346] In this implementation, the convolutional neural network-based in-loop filtering with adaptive model selection (DAM) is extended to the rate distortion optimization (RDO) process. And the number of residual blocks in DAM is reduced to 4. The DAM is applied to the coding unit level to select the best partitioning structure based on the RDO criterion. The rate distortion cost could be formulated as:

$$J = D + \lambda R$$

where D denotes the minimum value of distortion with DAM and without DAM.

[0347] Before applying the DAM, the cost J_A of partitioning mode A and the cost J_B of partitioning mode B are checked. When meeting the following condition, the DAM is skipped.

$$J_A > f_0 * J_B || J_B > f_1 * J_A$$

where the f_0 and f_1 are parameters.

5.2 Embodiment #2

5.2.1 Proposed Method

[0348] It is proposed that CNN-based filtering is involved during the partitioning mode selection. In particular, the samples obtained after CNN-based filtering are compared with original samples to calculate the distortion. The optimal partitioning mode is then selected based on the refined rate-distortion (RD) cost.

[0349] To reduce the complexity of applying CNN-based filtering in RDO, several fast algorithms are proposed. First, a simplified version of CNN model as shown in FIG. 16A and FIG. 16B is additionally trained and used in the RDO stage where the simplified model is implemented with SADL using fixed point-based calculation. Second, only one filter is included in the RDO process without considering filter selection. Finally, the proposed technique is only applied to the coding units with height and width no larger than 64. FIG. 16A illustrates an example diagram showing architecture of the proposed CNN filter, where M denotes the number of feature maps and N stands for the number of samples in one dimension. FIG. 16B illustrates an example diagram showing a construction of Attention Residual Block in FIG. 16A.

[0350] The inference and training processes of models are same as those in JVET-AA0111.

6.2.2 Inference

[0351] SADL is used for to perform the inference of the proposed CNN filters in RDO process. The network information in the inference stage is provided in Table 6.

TABLE 6

Network Information for NN-based Video Coding Tool Testing in Inference Stage Network Information in Inference Stage		
HW environment:		
Mandatory	GPU Type	N/A
	Framework:	SADL
	Number of GPUs per Task	0
	Total Parameter Number	1.56 M/model (4 models)
	Parameter Precision (Bits)	16 for fixed point version
Optional	Memory Parameter (MB)	3.1 MB/model (4 models)
	MAC (Giga)	539K/pixel
	Total Conv. Layers	25 + 16
	Total FC Layers	0
	Total Memory (MB)	
	Batch size:	1
	Patch size	128 × 128, 256 × 256
	Changes to network configuration or weights required to generate rate points	
	Peak Memory Usage	
	Other information:	

6.2.3 Training

[0352] PyTorch is used as the training platform. The DIV2K and BVI-DVC datasets are adopted to train the CNN filters of I slices and B slices, respectively. The network information in the training stage is provided in Table 7.

TABLE 7

Network Information for NN-based Video Coding Tool Testing in Training Stage Network Information in Training Stage		
Mandatory	GPU Type	GPU: Tesla-V100-SXM2-32GB
	Framework:	PyTorch v1.6
	Number of GPUs per Task	2
	Epoch:	50
	Batch size:	64
	Training time:	60h/model
	Training data information:	DIV2K, BVI-DVC
	Training configurations for generating compressed training data (if different to VTM CTC):	VTM-11.0 + new MCTE, QP { 17, 22, 27, 32, 37, 42}
	Loss function:	L1, L2
	Number of iterations	
Optional	Patch size	128 × 128
	Learning rate:	1e-4
	Optimizer:	ADAM
	Preprocessing:	
Other information:		

[0353] As used herein, the term “video unit” or “video block” may be a sequence, a picture, a slice, a tile, a brick, a subpicture, a coding tree unit (CTU)/coding tree block (CTB), a CTU/CTB row, one or multiple coding units (CUs)/coding blocks (CBs), one or multiple CTUs/CTBs, one or multiple Virtual Pipeline Data Unit (VPDU), a sub-region within a picture/slice/tile/brick. As used herein, the term “an independent filter (ID) filter” may refer to a filter is not exactly same with other filters and some parts of the filters are different, such as the input of the filter, the structure of the filter, the parameters of filter, the neural network model of the filter. In one example, the design of ID-Filter is unique and different with the design of other filters. In one example, the inputs of ID-Filter are different when filters share the consistent structure or consistent parameters or consistent model of neural network. ID-Filter can be any kind of filters, including filters without neural network (non-NN filter) and filters with neural network (NN filter). A Non-NN Filter may be one of deblocking filter (DF), sample adaptive offset (SAO), adaptive loop filter (ALF), etc. A NN filter can be any kind of NN filter, such as a convolutional neural network (CNN) filter. In the following discussion, a NN filter may also be referred to as a CNN filter.

[0354] FIG. 17 illustrates a flowchart of a method 1700 for video processing in accordance with embodiments of the present disclosure. The method 1700 is implemented during a conversion between a target video block of a video and a bitstream of the video.

[0355] At block 1710, for a conversion between a video unit of a video and a bitstream of the video unit, it is determined to whether to apply at least one neural network (NN) filter model or determine a rate distortion cost during a rate distortion optimization (RDO) process of the video unit based on at least one of: a distortion without NN filter model, a distortion with n-th NN filter model, a combination

of distortions of a plurality of NN filter models, or coding statistics of the video unit, and wherein n is an integer number.

[0356] At block 1720, a coding mode of the video unit is determined based on a rate distortion optimization (RDO) criterion in the RDO process.

[0357] At block 1730, the conversion is performed based on the coding mode. Alternatively, or in addition, the conversion may include decoding the video unit from the bitstream. In this way, the impact of reduce distortion due to NN filter is taken into consideration during the RDO process, thereby improving coding performances.

[0358] In some embodiments, the method 1700 further comprises: determining an approach to apply the at least one neural network (NN) filter model or determine the rate distortion cost during the RDO process of the video unit based on at least one of: the distortion without NN filter model, the distortion with n-th NN filter model, or the combination of distortions of the plurality of NN filter models.

[0359] In some embodiments, the distortion with the n-th NN filter model is derived by comparing filter reconstruction samples which are filtered by the n-th NN filter model with original samples. In some embodiments, the distortion in the RDO criterion is represented as $J=D+\lambda*R$, where D represents a distortion, λ represents a coefficient parameter, R represents a rate associated with a candidate coding mode.

[0360] In some embodiments, the distortion in the RDO criterion is the distortion with the n-th NN filter model. In some embodiments, the distortion in the RDO criterion is a minimal value of the distortion without NN filter model and the distortion with the n-th NN filter model. In some embodiments, n is one of: 1, 2, 3.

[0361] In some embodiments, the distortion in the RDO criterion is a distortion with a best NN filter model. In some embodiments, the distortion in the RDO criterion is a minimal value of distortion without NN filter model and a distortion with a best NN filter model.

[0362] In some embodiments, the best NN filter model is selected by distortion. In some embodiments, the best NN filter model is a default one.

[0363] In some embodiments, the distortion in the RDO criterion is the distortion without NN filter model multiplied a scaling factor. In some embodiments, the scaling factor is one of 1.0, 0.9, or 1.1.

[0364] In some embodiments, the distortion in the RDO criterion is derived according to the distortion without NN filter model and the distortion with the n-th NN filter model.

[0365] In some embodiments, the distortion in the RDO criterion is derived as $D=f_0*D_{ORG}+f_1*D''_{NNLF}$, and where D represents the distortion in the RDO criterion, f_0 represents a first scaling factor, f_1 represents a second scaling factor, D_{ORG} represents the distortion without NN filter model, and D''_{NNLF} represents the distortion with the n-th NN filter model. In some embodiments, $f_0=0.0$, $f_1=1.0$, or $f_0=1.0$, $f_1=0.0$.

[0366] In some embodiments, the distortion in the RDO criterion is derived as: $\min(f_0*D_{ORG}, f_1*D''_{NNLF})$, and where D represents the distortion in the RDO criterion, f_0 represents a first scaling factor, f_1 represents a second scaling factor, D_{ORG} represents the distortion without NN filter model, and D''_{NNLF} represents the distortion with the n-th NN filter model. In some embodiments, $f_0=1.0$, $f_1=1.0$.

[0367] In some embodiments, the distortion in the RDO criterion is a combination of one or more of: the distortion with the n-th NN filter model, a minimal value of the distortion without NN filter model and the distortion with the n-th NN filter model, a distortion with a best NN filter model, a minimal value of distortion without NN filter model and a distortion with a best NN filter model, the distortion without NN filter model multiplied a scaling factor, or a derivation according to the distortion without NN filter model and the distortion with the n-th NN filter model. In some embodiments, the combination is dependent on coding statistics of the video unit. In some embodiments, the combination is dependent on a type of candidate mode.

[0368] In some embodiments, if one or all of candidate modes are not partitioning mode, the distortion in the RDO criterion is a combination of at least one of: a distortion with p-th NN filter model or the distortion without NN filter model, where p is an integer number. In some embodiments, if the p-th NN filter model is not applied for the one or all of candidate modes, the distortion in the RDO criterion is the distortion without NN filter model. In some embodiments, if the p-th NN filter model is applied for the one or all of candidate modes, the distortion in the RDO criterion is the distortion with the p-th NN filter model. In some embodiments, if one or all of candidate modes are partitioning modes, the distortion in the RDO criterion is a combination of at least one of: a distortion with q-th NN filter model, a distortion with p-th NN filter model, or the distortion without NN filter model, wherein p and q are integer numbers.

[0369] In some embodiments, the combination is dependent on at least one of: usage of q-th NN filter model or usage of p-th NN filter model, where p and q are integer numbers. In some embodiments, if the q-th NN filter model is applied for one or all of candidate modes, the distortion in the RDO criterion is the minimal value of the distortion without NN filter model and distortion with the q-th NN filter model. In some embodiments, if the q-th NN filter model is not applied for one or all of candidate modes, the distortion in the RDO criterion is a combination of the distortion with the p-th NN filter model and the distortion without NN filter model. In some embodiments, if the p-th NN filter model is applied for the one or all of candidate modes, the distortion in the RDO criterion is the distortion with the p-th NN filter model. In some embodiments, if the p-th NN filter model is not applied for the one or all of candidate modes, the distortion in the RDO criterion is the distortion without NN filter model.

[0370] In some embodiments, an approach of determining the distortion in the RDO criterion is applied according to a predefined order or an adaptive order. In some embodiments, the order of applying the approach of determining the distortion in the RDO criterion is dependent on at least one of: a coding mode of the video unit or coding statistics of the video unit.

[0371] In some embodiments, an approach of determining the distortion in the RDO criterion as the minimal value of the distortion without NN filter model and distortion with the q-th NN filter model has a higher priority than an approach of determining the distortion in the RDO criterion as the distortion with the p-th NN filter model. In some embodiments, an approach of determining the distortion in the RDO criterion as the distortion with the p-th NN filter

model has a higher priority than an approach of determining the distortion in the RDO criterion as the distortion without NN filter model.

[0372] In some embodiments, the p-th NN filter model is one of: a deblocking filter, a sample adaptive offset (SAO) filter, or an adaptive loop filter (ALF). In some embodiments, the q-th NN filter model is a convolutional neural network (CNN) filter. In some embodiments, the q-th NN filter model is the same with the p-th NN filter model. In some embodiments, p is equal to q.

[0373] In some embodiments, a number of candidate modes are comparable modes with the RDO criterion which is represented as $J=D+\lambda \cdot R$. In some embodiments, the number candidate modes is equal to 1 or 2 or 3 or 4.

[0374] In some embodiments, at least one of: f , f_0 or f_1 is set according to at least one of: a temporal layer or a slice type. In some embodiments, at least one of: f , f_0 or f_1 is set according to quantization parameter (QP). In some embodiments, at least one of: f , f_0 or f_1 is set according to a configuration associated with the video unit. In some embodiments, a same rule of usage of NN filtering is applied to determine the rate distortion cost for at least one of: all candidate modes or partitioning approaches for the video unit.

[0375] In some embodiments, the distortion in the RDO criterion is the minimal value of distortions with m NN filter models, where m is an integer number. In some embodiments, the distortion in the RDO criterion is a minimal value of the distortion without NN filter model and distortions with m NN filter models, where m is an integer number.

[0376] In some embodiments, m is an available number of constructed NN filter models. In some embodiments, m is dependent on a type of video unit. In some embodiments, m is dependent on a signaled parameter of video unit. In some embodiments, m is a default value. In some embodiments, m is one of: 0, 1, 2, 3, or 4.

[0377] In some embodiments, the distortion in the RDO criterion is a combination of the distortion without NN filter model and distortions with m NN filter models, where m is an integer number. In some embodiments, the distortion in the RDO criterion is derived as: $D=f_0 \cdot D_{ORG} + \sum f_n \cdot D_{NNLF}^n$, where D represents the distortion in the RDO criterion, f_0 represents a first scaling factor, f_1 represents a second scaling factor, D_{ORG} represents the distortion without NN filter model, and D_{NNLF}^n represents the distortion with the n-th NN filter model.

[0378] In some embodiments, at least one of: m, f_0 or f_n is set according to at least one of: a temporal layer or a slice type. In some embodiments, at least one of: m, f_0 or f_n is set according to quantization parameter (QP). In some embodiments, a same rule of usage of NN filtering is applied to determine the rate distortion cost for at least one of: all candidate modes or partitioning approaches for the video unit.

[0379] In some embodiments, the distortion without NN filter model is dependent on another filter which is different with the at least one NN filter model. In some embodiments, the distortion with the other filter is derived by comparing filter reconstruction samples which are filtered by the other filter with original samples. In some embodiments, the other filter is applied on reconstruction samples before the at least one NN filter model.

[0380] In some embodiments, the at least one NN filter model is applied after the other filter. In some embodiments,

the distortion without NN filter model is the distortion with the other filter multiplied a scaling factor. In some embodiments, the distortion without NN filter model is a combination of the distortions with a plurality of other filters.

[0381] In some embodiments, an input of the at least one NN filter model in the RDO process include a signal from at least one of: a current block or a neighboring block. In some embodiments, at least one of the followings from the current block is involved in a NN filter process: a prediction signal, partitioning information, or a reconstruction signal.

[0382] In some embodiments, other information from one or more reference frames is involved in a NN filter process of the video unit. In some embodiments, at least one of: a collocated block from a first frame in list-0, or a collocated block from a first frame in list-1 is involved in the NN filter process. In some embodiments, one or more motion compensated reference blocks are involved in the NN filter process.

[0383] In some embodiments, at least one of the following from at least one neighboring block is involved in a NN filter process of the video unit: a prediction signal, partitioning information, or pixels. In some embodiments, the at least one neighboring block is located at one of: a left side of the video unit, an above side of the video unit, or a left-above side of the video unit. In some embodiments, samples of the at least one neighboring block are reconstructed. In some embodiments, the at least one neighboring block is located at a right side or a below side of the video unit.

[0384] In some embodiments, samples of neighboring blocks are no reconstructed. In some embodiments, samples of neighboring blocks are an original signal. In some embodiments, samples of neighboring blocks are derived from the video unit.

[0385] In some embodiments, the coding statistics comprises at least one of: a prediction mode, QP, a temporal layer, a slice type, or a dimension of the video unit.

[0386] In some embodiments, if a width of the video unit is less than or equal to a first threshold and a height of the video unit is less than or equal to a second threshold, the at least one NN filter model is applied in the RDO process. In some embodiments, the first threshold is 64 and the second threshold is 64.

[0387] In some embodiments, if a width of the video unit is larger than or equal to a first threshold and a height of the video unit is larger than or equal to a second threshold, the at least one NN filter model is applied in the RDO process. In some embodiments, the first threshold is 16 and the second threshold is 16.

[0388] In some embodiments, if a value of the width of the video unit multiplying the height of the video unit is larger than or equal to a third threshold, the at least one NN filter model is applied in the RDO process. In some embodiments, the third threshold is 64.

[0389] In some embodiments, if a value of the width of the video unit multiplying the height of the video unit is less than or equal to a fourth threshold, the at least one NN filter model is applied in the RDO process. In some embodiments, the fourth threshold is 4094.

[0390] In some embodiments, the method 1700 further comprises: determining whether to and/or the approach to apply the at least one NN filter in the RDO process based on color components. In some embodiments, the at least one NN filter model is only applied in a certain color component. In some embodiments, the certain component is Luma

component or Chroma component. In some embodiments, the at least one NN filter model is applied for all color components.

[0391] In some embodiments, the method 1700 further comprises: determining whether to and/or the approach to apply the at least one NN filter in the RDO process based on a rate distortion cost without NN filter model.

[0392] In some embodiments, if a first rate distortion cost of a first coding mode is higher than or equal to a second rate distortion cost of a second coding mode multiplied a first factor, the at least one NN filter is not applied. In some embodiments, the first factor is one of: 1.0, 1.001, 1.005, 1.05, or 1.01.

[0393] In some embodiments, if a second rate distortion cost of a second coding mode is higher than or equal to a first rate distortion cost of a first coding mode multiplied a second factor, the at least one NN filter is not applied. In some embodiments, the second factor is one of: 1.0, 1.001, 1.005, 1.05, or 1.01. In some embodiments, if a first rate distortion cost of a first coding mode is higher than or equal to a second rate distortion cost of a second coding mode multiplied a first factor, or if a second rate distortion cost of a second coding mode is higher than or equal to a first rate distortion cost of a first coding mode multiplied a second factor, the at least one NN filter is not applied.

[0394] In some embodiments, if a ratio between a first rate distortion cost of a first coding mode and a second rate distortion cost of a second coding mode is higher than or equal to a first ratio threshold, the at least one NN filter model is not applied. In some embodiments, the first ratio threshold is one of: 1.0, 1.001, 1.005, 1.05, or 1.01.

[0395] In some embodiments, if a ratio between a first rate distortion cost of a first coding mode and a second rate distortion cost of a second coding mode is less than or equal to a second ratio threshold, the at least one NN filter model is not applied. In some embodiments, the second ratio threshold is one of: 1.0, 1.001, 1.005, 1.05, or 1.01. In some embodiments, if a ratio between a first rate distortion cost of a first coding mode and a second rate distortion cost of a second coding mode is higher than or equal to a first ratio threshold or if a ratio between a first rate distortion cost of a first coding mode and a second rate distortion cost of a second coding mode is less than or equal to a second ratio threshold, the at least one NN filter is not applied.

[0396] In some embodiments, the at least one NN filter is not applied, if one of the followings is satisfied: a first rate distortion cost of a first coding mode is equal to a first cost threshold and a second rate distortion of a second coding mode is equal to a second cost threshold, the first rate distortion cost is smaller than the first cost threshold and the second rate distortion is smaller than the second cost threshold, or the first rate distortion cost is higher than the first cost threshold and the second rate distortion is higher than the second cost threshold. In some embodiments, the at least one NN filter is applied, if one of the followings is satisfied: a first rate distortion cost of a first coding mode is equal to a first cost threshold and a second rate distortion of a second coding mode is equal to a second cost threshold, the first rate distortion cost is smaller than the first cost threshold and the second rate distortion is smaller than the second cost threshold, or the first rate distortion cost is higher than the first cost threshold and the second rate distortion is higher than the second cost threshold.

[0397] In some embodiments, the first cost threshold or the second cost threshold is equal to MAX_DOUBLE which is a default value. In some embodiments, the first cost threshold or the second cost threshold is equal to 1.7e+308.

[0398] In some embodiments, the at least one NN filter for a first coding mode and a second coding mode is applied according to a predefined order or an adaptive order. In some embodiments, an order of applying the at least one NN filter for the first coding mode and the second coding mode is dependent on at least one of: a coding mode of the video unit and coding statistics of the video unit. In some embodiments, an order of applying the at least one NN filter for the first coding mode and the second coding mode is dependent on a first rate distortion cost of a first coding mode and a second rate distortion of a second coding mode.

[0399] In some embodiments, if the first rate distortion cost of the first coding mode is higher than or equal to the second rate distortion cost of the second coding mode, applying the at least one NN filter to the first coding mode has a higher priority than applying the at least one NN filter to the second coding mode. In some embodiments, if the second rate distortion cost of the second coding mode is higher than or equal to the first rate distortion cost of the first coding mode, applying the at least one NN filter to the first coding mode has a higher priority than applying the at least one NN filter to the second coding mode.

[0400] In some embodiments, a first rate distortion cost of a first coding mode is equal to the RDO criterion in the RDO process. In some embodiments, a second rate distortion cost of a second coding mode is equal to the RDO criterion in the RDO process. In some embodiments, the distortion in the RDO criterion is equal to the distortion without NN filter model.

[0401] In some embodiments, at least one of the first threshold, the second threshold, the third threshold, the fourth threshold, the first cost threshold, or the second cost threshold is set according to temporal layers. In some embodiments, at least one of the first threshold, the second threshold, the third threshold, the fourth threshold, the first cost threshold, or the second cost threshold is set according to the QP. In some embodiments, at least one of the first threshold, the second threshold, the third threshold, the fourth threshold, the first cost threshold, or the second cost threshold is set according to a slice type. In some embodiments, at least one of the first threshold, the second threshold, the third threshold, the fourth threshold, the first cost threshold, or the second cost threshold is set according to a configuration of the video unit.

[0402] In some embodiments, the method 1700 further comprises: determining whether to and/or the approach to apply the at least one NN filter in the RDO process based on a rate distortion cost without NN filter model and a rate distortion cost with NN filter.

[0403] In some embodiments, whether to apply the at least one NN filter model for a first coding mode is dependent on a rate distortion cost of the first coding mode without NN filter model and a rate distortion cost of a second coding mode with NN filter model. In some embodiments, a rate distortion cost of the first coding mode without NN filter model is equal to the RDO criterion in the RDO process. In some embodiments, the distortion in the RDO criterion is equal to the distortion without NN filter model. In some

embodiments, a rate distortion cost of a second coding mode with NN filter model is equal to the RDO criterion in the RDO process.

[0404] In some embodiments, the distortion in the RDO criterion is equal to a minimal value of the distortion without NN filter model and the distortion with the n-th NN filter model. In some embodiments, the distortion in the RDO criterion is equal to the distortion with the n-th NN filter model.

[0405] In some embodiments, if a rate distortion cost of the first coding mode without NN filter model is higher than or equal to a rate distortion cost of a second coding mode with NN filter model multiplied a first scaling factor, the at least one NN filter is not applied for the first coding mode. In some embodiments, if a rate distortion cost of a second coding mode with NN filter model multiplied a second scaling factor is higher than or equal to a rate distortion cost of the first coding mode without NN filter model, the at least one NN filter is not applied for the first coding mode. In some embodiments, if a rate distortion cost of the first coding mode without NN filter model is higher than or equal to a rate distortion cost of a second coding mode with NN filter model multiplied a first scaling factor, or if the rate distortion cost of the second coding mode with NN filter model multiplied a second scaling factor is higher than or equal to the rate distortion cost of the first coding mode without NN filter model, the at least one NN filter is not applied for the first coding mode.

[0406] In some embodiments, if a ratio between a rate distortion cost of the first coding mode without NN filter model and a rate distortion cost of a second coding mode with NN filter model is higher than or equal to a first ratio threshold, the at least one NN filter model is not applied. In some embodiments, if a ratio between a rate distortion cost of the first coding mode without NN filter model and a rate distortion cost of a second coding mode with NN filter model is less than or equal to a second ratio threshold, the at least one NN filter model is not applied. In some embodiments, if a ratio between a rate distortion cost of the first coding mode without NN filter model and a rate distortion cost of a second coding mode with NN filter model is higher than or equal to a first ratio threshold, or if a ratio between the rate distortion cost of the first coding mode without NN filter model and the rate distortion cost of the second coding mode with NN filter model is less than or equal to a second ratio threshold, the at least one NN filter is not applied.

[0407] In some embodiments, at least one of: the first scaling factor, the second scaling factor, the first ratio threshold, or the second ratio threshold is set according to temporal layers. In some embodiments, at least one of: the first scaling factor, the second scaling factor, the first ratio threshold, or the second ratio threshold is set according to the QP. In some embodiments, at least one of: the first scaling factor, the second scaling factor, the first ratio threshold, or the second ratio threshold is set according to a slice type. In some embodiments, at least one of: the first scaling factor, the second scaling factor, the first ratio threshold, or the second ratio threshold is set according to a configuration of the video unit.

[0408] In some embodiments, the at least one NN filter is only applied to partial samples of a block in the RDO process. In some embodiments, the at least one NN filter is only applied to a center subblock in the block. In some embodiments, a width of the subblock is equal to half of

width of the block, and a height of the subblock is equal to half of height of the block. In some embodiments, a width of the subblock is equal to three quarters of width of the block, and a height of the subblock is equal to three quarters of height of the block.

[0409] In some embodiments, the method **1700** further comprises: determining whether to and/or the approach to apply the at least one NN filter in the RDO process based on temporal layers. In some embodiments, the at least one NN filter is applied to the temporal layer with ID which is greater than K. In some embodiments, the at least one NN filter is applied to the temporal layer with ID which is smaller than K, where K is an integer number. In some embodiments, K is equal to one of 0, 1, 2, 3, 4, 5, or 6.

[0410] In some embodiments, the method **1700** further comprises: determining whether to and/or the approach to apply the at least one NN filter in the RDO process based on coding statistics of sub coding units. In some embodiments, if the at least one NN filter is applied to a portion or all of the sub coding units, the at least one NN filter is not applied to the video unit. In some embodiments, if the at least one NN filter is applied to the portion or all of the sub coding units, the at least one NN filter is applied to the video unit.

[0411] In some embodiments, if at least one of: rate-distortion costs or distortions of the portion or all of the sub coding units are available, the at least one NN filter is not applied to the video unit. In some embodiments, if at least one of: rate-distortion costs or distortions of the portion or all of the sub coding units are available, the at least one NN filter is applied to the video unit.

[0412] According to further embodiments of the present disclosure, a non-transitory computer-readable recording medium is provided. The non-transitory computer-readable recording medium stores a bitstream of a video which is generated by a method performed by an apparatus for video processing. The method comprises: determining whether to apply at least one neural network (NN) filter model or determine a rate distortion cost during a rate distortion optimization (RDO) process of a video unit of the video based on at least one of: a distortion without NN filter model, a distortion with n-th NN filter model, a combination of distortions of a plurality of NN filter models, or coding statistics of the video unit, and wherein n is an integer number; determining a coding mode of the video unit based on a rate distortion optimization (RDO) criterion in the RDO process; and generating the bitstream based on the coding mode.

[0413] According to still further embodiments of the present disclosure, a method for storing bitstream of a video is provided. The method comprises: determining whether to apply at least one neural network (NN) filter model or determine a rate distortion cost during a rate distortion optimization (RDO) process of a video unit of the video based on at least one of: a distortion without NN filter model, a distortion with n-th NN filter model, a combination of distortions of a plurality of NN filter models, or coding statistics of the video unit, and wherein n is an integer number; determining a coding mode of the video unit based on a rate distortion optimization (RDO) criterion in the RDO process; generating the bitstream based on the coding mode; and storing the bitstream in a non-transitory computer-readable recording medium.

[0414] Implementations of the present disclosure can be described in view of the following clauses, the features of which can be combined in any reasonable manner.

[0415] Clause 1. A method of video processing, comprising: determining, for a conversion between a video unit of a video and a bitstream of the video unit, whether to apply at least one neural network (NN) filter model or determine a rate distortion cost during a rate distortion optimization (RDO) process of the video unit based on at least one of: a distortion without NN filter model, a distortion with n-th NN filter model, a combination of distortions of a plurality of NN filter models, or coding statistics of the video unit, and wherein n is an integer number; determining a coding mode of the video unit based on a rate distortion optimization (RDO) criterion in the RDO process; and performing the conversion based on the coding mode.

[0416] Clause 2. The method of clause 1, further comprising: determining an approach to apply the at least one neural network (NN) filter model or determine the rate distortion cost during the RDO process of the video unit based on at least one of: the distortion without NN filter model, the distortion with n-th NN filter model, or the combination of distortions of the plurality of NN filter models.

[0417] Clause 3. The method of clause 1 or 2, wherein the distortion with the n-th NN filter model is derived by comparing filter reconstruction samples which are filtered by the n-th NN filter model with original samples, and wherein the distortion in the RDO criterion is represented as $J=D+\lambda R$, wherein D represents a distortion, λ represents a coefficient parameter, R represents a rate associated with a candidate coding mode.

[0418] Clause 4. The method of any of clauses 1-3, wherein the distortion in the RDO criterion is the distortion with the n-th NN filter model.

[0419] Clause 5. The method of any of clauses 1-3, wherein the distortion in the RDO criterion is a minimal value of the distortion without NN filter model and the distortion with the n-th NN filter model.

[0420] Clause 6. The method of clause 4 or 5, wherein n is one of: 1, 2, 3.

[0421] Clause 7. The method of any of clauses 1-3, wherein the distortion in the RDO criterion is a distortion with a best NN filter model.

[0422] Clause 8. The method of any of clauses 1-3, wherein the distortion in the RDO criterion is a minimal value of distortion without NN filter model and a distortion with a best NN filter model.

[0423] Clause 9. The method of clause 7, wherein the best NN filter model is selected by distortion, or wherein the best NN filter model is a default one.

[0424] Clause 10. The method of any of clauses 1-3, wherein the distortion in the RDO criterion is the distortion without NN filter model multiplied a scaling factor.

[0425] Clause 11. The method of clause 10, wherein the scaling factor is one of 1.0, 0.9, or 1.1.

[0426] Clause 12. The method of any of clauses 1-3, wherein the distortion in the RDO criterion is derived according to the distortion without NN filter model and the distortion with the n-th NN filter model.

[0427] Clause 13. The method of clause 12, wherein the distortion in the RDO criterion is derived as: $D=f_0*D_{ORG}+f_1*D''_{NNLF}$, and wherein D represents the distortion in the RDO criterion, f_0 represents a first scaling factor, f_1 represents a second scaling factor, D_{ORG} represents the distortion

without NN filter model, and D''_{NNLF} represents the distortion with the n-th NN filter model.

[0428] Clause 14. The method of clause 13, wherein $f_0=0.0$, $f_1=1.0$, or wherein $f_0=1.0$, $f_1=0.0$.

[0429] Clause 15. The method of clause 12, wherein the distortion in the RDO criterion is derived as: $\min(f_0*D_{ORG}, f_1*D''_{NNLF})$, and wherein D represents the distortion in the RDO criterion, f_0 represents a first scaling factor, f_1 represents a second scaling factor, D_{ORG} represents the distortion without NN filter model, and D''_{NNLF} represents the distortion with the n-th NN filter model.

[0430] Clause 16. The method of clause 15, wherein $f_0=1.0$, $f_1=1.0$.

[0431] Clause 17. The method of any of clauses 1-3, wherein the distortion in the RDO criterion is a combination of one or more of: the distortion with the n-th NN filter model, a minimal value of the distortion without NN filter model and the distortion with the n-th NN filter model, a distortion with a best NN filter model, a minimal value of distortion without NN filter model and a distortion with a best NN filter model, the distortion without NN filter model multiplied a scaling factor, or a derivation according to the distortion without NN filter model and the distortion with the n-th NN filter model.

[0432] Clause 18. The method of clause 17, wherein the combination is dependent on coding statistics of the video unit.

[0433] Clause 19. The method of clause 18, wherein the combination is dependent on a type of candidate mode.

[0434] Clause 20. The method of clause 17, wherein if one or all of candidate modes are not partitioning mode, the distortion in the RDO criterion is a combination of at least one of: a distortion with p-th NN filter model or the distortion without NN filter model, wherein p is an integer number.

[0435] Clause 21. The method of clause 20, wherein if the p-th NN filter model is not applied for the one or all of candidate modes, the distortion in the RDO criterion is the distortion without NN filter model.

[0436] Clause 22. The method of clause 20, wherein if the p-th NN filter model is applied for the one or all of candidate modes, the distortion in the RDO criterion is the distortion with the p-th NN filter model.

[0437] Clause 23. The method of clause 17, wherein if one or all of candidate modes are partitioning modes, the distortion in the RDO criterion is a combination of at least one of: a distortion with q-th NN filter model, a distortion with p-th NN filter model, or the distortion without NN filter model, wherein p and q are integer numbers.

[0438] Clause 24. The method of clause 17, wherein the combination is dependent on at least one of: usage of q-th NN filter model or usage of p-th NN filter model, wherein p and q are integer numbers.

[0439] Clause 25. The method of clause 24, wherein if the q-th NN filter model is applied for one or all of candidate modes, the distortion in the RDO criterion is the minimal value of the distortion without NN filter model and distortion with the q-th NN filter model.

[0440] Clause 26. The method of clause 24, wherein if the q-th NN filter model is not applied for one or all of candidate modes, the distortion in the RDO criterion is a combination of the distortion with the p-th NN filter model and the distortion without NN filter model.

[0441] Clause 27. The method of clause 26, wherein if the p-th NN filter model is applied for the one or all of candidate modes, the distortion in the RDO criterion is the distortion with the p-th NN filter model.

[0442] Clause 28. The method of clause 26, wherein if the p-th NN filter model is not applied for the one or all of candidate modes, the distortion in the RDO criterion is the distortion without NN filter model.

[0443] Clause 29. The method of clause 17, wherein an approach of determining the distortion in the RDO criterion is applied according to a predefined order or an adaptive order.

[0444] Clause 30. The method of clause 29, wherein the order of applying the approach of determining the distortion in the RDO criterion is dependent on at least one of: a coding mode of the video unit or coding statistics of the video unit.

[0445] Clause 31. The method of clause 29, wherein an approach of determining the distortion in the RDO criterion as the minimal value of the distortion without NN filter model and distortion with the q-th NN filter model has a higher priority than an approach of determining the distortion in the RDO criterion as the distortion with the p-th NN filter model.

[0446] Clause 32. The method of clause 29, wherein an approach of determining the distortion in the RDO criterion as the distortion with the p-th NN filter model has a higher priority than an approach of determining the distortion in the RDO criterion as the distortion without NN filter model.

[0447] Clause 33. The method of any of clauses 17-32, wherein the p-th NN filter model is one of: a deblocking filter, a sample adaptive offset (SAO) filter, or an adaptive loop filter (ALF).

[0448] Clause 34. The method of any of clauses 17-32, wherein the q-th NN filter model is a convolutional neural network (CNN) filter.

[0449] Clause 35. The method of any of clauses 17-32, wherein the q-th NN filter model is the same with the p-th NN filter model.

[0450] Clause 36. The method of clause 15, wherein p is equal to q.

[0451] Clause 37. The method of clause 17, wherein a number of candidate modes are comparable modes with the RDO criterion which is represented as $J=D+\lambda R$.

[0452] Clause 38. The method of clause 17, wherein the number candidate modes is equal to 1 or 2 or 3 or 4.

[0453] Clause 39. The method of any of clauses 1-38, wherein at least one of: f , f_0 or f_1 is set according to at least one of: a temporal layer or a slice type.

[0454] Clause 40. The method of any of clauses 1-38, wherein at least one of: f , f_0 or f_1 is set according to quantization parameter (QP).

[0455] Clause 41. The method of any of clauses 1-38, wherein at least one of: f , f_0 or f_1 is set according to a configuration associated with the video unit.

[0456] Clause 42. The method of any of clauses 1-41, wherein a same rule of usage of NN filtering is applied to determine the rate distortion cost for at least one of: all candidate modes or partitioning approaches for the video unit.

[0457] Clause 43. The method of any of clauses 1-3, wherein the distortion in the RDO criterion is the minimal value of distortions with m NN filter models, wherein m is an integer number, or wherein the distortion in the RDO criterion is a minimal value of the distortion without NN

filter model and distortions with m NN filter models, wherein m is an integer number.

[0458] Clause 44. The method of clause 43, wherein m is an available number of constructed NN filter models.

[0459] Clause 45. The method of clause 43, wherein m is dependent on a type of video unit.

[0460] Clause 46. The method of clause 43, wherein m is dependent on a signaled parameter of video unit.

[0461] Clause 47. The method of clause 43, wherein m is a default value.

[0462] Clause 48. The method of clause 43, wherein m is one of: 0, 1, 2, 3, or 4.

[0463] Clause 49. The method of any of clauses 1-3, wherein the distortion in the RDO criterion is a combination of the distortion without NN filter model and distortions with m NN filter models, wherein m is an integer number.

[0464] Clause 50. The method of clause 49, wherein the distortion in the RDO criterion is derived as: $D=f_0 \cdot D_{ORG} + \sum f_n \cdot D_{NNLF}^n$, and wherein D represents the distortion in the RDO criterion, f_0 represents a first scaling factor, f_1 represents a second scaling factor, D_{ORG} represents the distortion without NN filter model, and D_{NNLF}^n represents the distortion with the n-th NN filter model.

[0465] Clause 51. The method of any of clauses 43-50, wherein at least one of: m, f_0 or f_n is set according to at least one of: a temporal layer or a slice type.

[0466] Clause 52. The method of any of clauses 43-50, wherein at least one of: m, f_0 or f_n is set according to quantization parameter (QP).

[0467] Clause 53. The method of any of clauses 43-52, wherein a same rule of usage of NN filtering is applied to determine the rate distortion cost for at least one of: all candidate modes or partitioning approaches for the video unit.

[0468] Clause 54. The method of any of clauses 1-3, wherein the distortion without NN filter model is dependent on another filter which is different with the at least one NN filter model.

[0469] Clause 55. The method of clause 54, wherein the distortion with the other filter is derived by comparing filter reconstruction samples which are filtered by the other filter with original samples.

[0470] Clause 56. The method of clause 54, wherein the other filter is applied on reconstruction samples before the at least one NN filter model.

[0471] Clause 57. The method of clause 54, wherein the at least one NN filter model is applied after the other filter.

[0472] Clause 58. The method of clause 54, wherein the distortion without NN filter model is the distortion with the other filter multiplied a scaling factor.

[0473] Clause 59. The method of clause 54, wherein the distortion without NN filter model is a combination of the distortions with a plurality of other filters.

[0474] Clause 60. The method of any of clauses 1-3, wherein an input of the at least one NN filter model in the RDO process include a signal from at least one of: a current block or a neighboring block.

[0475] Clause 61. The method of clause 60, wherein at least one of the followings from the current block is involved in a NN filter process: a prediction signal, partitioning information, or a reconstruction signal.

[0476] Clause 62. The method of clause 60 or 61, wherein other information from one or more reference frames is involved in a NN filter process of the video unit.

[0477] Clause 63. The method of clause 62, wherein at least one of: a collocated block from a first frame in list-0, or a collocated block from a first frame in list-1 is involved in the NN filter process.

[0478] Clause 64. The method of clause 62, wherein one or more motion compensated reference blocks are involved in the NN filter process.

[0479] Clause 65. The method of any of clauses 60-64, wherein at least one of the following from at least one neighboring block is involved in a NN filter process of the video unit: a prediction signal, partitioning information, or pixels.

[0480] Clause 66. The method of clause 65, wherein the at least one neighboring block is located at one of: a left side of the video unit, an above side of the video unit, or a left-above side of the video unit.

[0481] Clause 67. The method of clause 65, wherein samples of the at least one neighboring block are reconstructed.

[0482] Clause 68. The method of clause 65, wherein the at least one neighboring block is located at a right side or a below side of the video unit.

[0483] Clause 69. The method of clause 65, wherein samples of neighboring blocks are not reconstructed.

[0484] Clause 70. The method of clause 65, wherein samples of neighboring blocks are an original signal.

[0485] Clause 71. The method of clause 65, wherein samples of neighboring blocks are derived from the video unit.

[0486] Clause 72. The method of any of clauses 1-3, wherein the coding statistics comprises at least one of: a prediction mode, QP, a temporal layer, a slice type, or a dimension of the video unit.

[0487] Clause 73. The method of any of clauses 1-3, wherein if a width of the video unit is less than or equal to a first threshold and a height of the video unit is less than or equal to a second threshold, the at least one NN filter model is applied in the RDO process.

[0488] Clause 74. The method of clause 73, wherein the first threshold is 64 and the second threshold is 64.

[0489] Clause 75. The method of any of clauses 1-3, wherein if a width of the video unit is larger than or equal to a first threshold and a height of the video unit is larger than or equal to a second threshold, the at least one NN filter model is applied in the RDO process.

[0490] Clause 76. The method of clause 75, wherein the first threshold is 16 and the second threshold is 16.

[0491] Clause 77. The method of any of clauses 1-3, wherein if a value of the width of the video unit multiplying the height of the video unit is larger than or equal to a third threshold, the at least one NN filter model is applied in the RDO process.

[0492] Clause 78. The method of clause 77, wherein the third threshold is 64.

[0493] Clause 79. The method of any of clauses 1-3, wherein if a value of the width of the video unit multiplying the height of the video unit is less than or equal to a fourth threshold, the at least one NN filter model is applied in the RDO process.

[0494] Clause 80. The method of clause 79, wherein the fourth threshold is 4094.

[0495] Clause 81. The method of any of clauses 1-3, further comprising: determining whether to and/or the

approach to apply the at least one NN filter in the RDO process based on color components.

[0496] Clause 82. The method of clause 81, wherein the at least one NN filter model is only applied in a certain color component.

[0497] Clause 83. The method of clause 82, wherein the certain component is Luma component or Chroma component.

[0498] Clause 84. The method of clause 81, wherein the at least one NN filter model is applied for all color components.

[0499] Clause 85. The method of any of clauses 1-3, further comprising: determining whether to and/or the approach to apply the at least one NN filter in the RDO process based on a rate distortion cost without NN filter model.

[0500] Clause 86. The method of clause 85, wherein if a first rate distortion cost of a first coding mode is higher than or equal to a second rate distortion cost of a second coding mode multiplied a first factor, the at least one NN filter is not applied.

[0501] Clause 87. The method of clause 86, wherein the first factor is one of: 1.0, 1.001, 1.005, 1.05, or 1.01.

[0502] Clause 88. The method of any of clauses 85-87, wherein if a second rate distortion cost of a second coding mode is higher than or equal to a first rate distortion cost of a first coding mode multiplied a second factor, the at least one NN filter is not applied.

[0503] Clause 89. The method of clause 88, wherein the second factor is one of: 1.0, 1.001, 1.005, 1.05, or 1.01.

[0504] Clause 90. The method of any of clauses 85-89, wherein if a first rate distortion cost of a first coding mode is higher than or equal to a second rate distortion cost of a second coding mode multiplied a first factor, or if a second rate distortion cost of a second coding mode is higher than or equal to a first rate distortion cost of a first coding mode multiplied a second factor, the at least one NN filter is not applied.

[0505] Clause 91. The method of any of clauses 85-90, wherein if a ratio between a first rate distortion cost of a first coding mode and a second rate distortion cost of a second coding mode is higher than or equal to a first ratio threshold, the at least one NN filter model is not applied.

[0506] Clause 92. The method of clause 91, wherein the first ratio threshold is one of: 1.0, 1.001, 1.005, 1.05, or 1.01.

[0507] Clause 93. The method of any of clauses 85-92, wherein if a ratio between a first rate distortion cost of a first coding mode and a second rate distortion cost of a second coding mode is less than or equal to a second ratio threshold, the at least one NN filter model is not applied.

[0508] Clause 94. The method of clause 93, wherein the second ratio threshold is one of: 1.0, 1.001, 1.005, 1.05, or 1.01.

[0509] Clause 95. The method of any of clauses 85-94, wherein if a ratio between a first rate distortion cost of a first coding mode and a second rate distortion cost of a second coding mode is higher than or equal to a first ratio threshold or if a ratio between a first rate distortion cost of a first coding mode and a second rate distortion cost of a second coding mode is less than or equal to a second ratio threshold, the at least one NN filter is not applied.

[0510] Clause 96. The method of any of clauses 85-95, wherein the at least one NN filter is not applied, if one of the followings is satisfied: a first rate distortion cost of a first

coding mode is equal to a first cost threshold and a second rate distortion of a second coding mode is equal to a second cost threshold, the first rate distortion cost is smaller than the first cost threshold and the second rate distortion is smaller than the second cost threshold, or the first rate distortion cost is higher than the first cost threshold and the second rate distortion is higher than the second cost threshold.

[0511] Clause 97. The method of any of clauses 85-95, wherein the at least one NN filter is applied, if one of the followings is satisfied: a first rate distortion cost of a first coding mode is equal to a first cost threshold and a second rate distortion of a second coding mode is equal to a second cost threshold, the first rate distortion cost is smaller than the first cost threshold and the second rate distortion is smaller than the second cost threshold, or the first rate distortion cost is higher than the first cost threshold and the second rate distortion is higher than the second cost threshold.

[0512] Clause 98. The method of clause 96 or 97, wherein the first cost threshold or the second cost threshold is equal to MAX_DOUBLE which is a default value, or wherein the first cost threshold or the second cost threshold is equal to $1.7e+308$.

[0513] Clause 99. The method of any of clauses 85-98, wherein the at least one NN filter for a first coding mode and a second coding mode is applied according to a predefined order or an adaptive order.

[0514] Clause 100. The method of clause 99, wherein an order of applying the at least one NN filter for the first coding mode and the second coding mode is dependent on at least one of: a coding mode of the video unit and coding statistics of the video unit.

[0515] Clause 101. The method of clause 99, wherein an order of applying the at least one NN filter for the first coding mode and the second coding mode is dependent on a first rate distortion cost of a first coding mode and a second rate distortion of a second coding mode.

[0516] Clause 102. The method of clause 101, wherein if the first rate distortion cost of the first coding mode is higher than or equal to the second rate distortion cost of the second coding mode, applying the at least one NN filter to the first coding mode has a higher priority than applying the at least one NN filter to the second coding mode.

[0517] Clause 103. The method of clause 101, wherein if the second rate distortion cost of the second coding mode is higher than or equal to the first rate distortion cost of the first coding mode, applying the at least one NN filter to the first coding mode has a higher priority than applying the at least one NN filter to the second coding mode.

[0518] Clause 104. The method of any of clauses 85-103, wherein a first rate distortion cost of a first coding mode is equal to the RDO criterion in the RDO process.

[0519] Clause 105. The method of any of clauses 85-104, wherein a second rate distortion cost of a second coding mode is equal to the RDO criterion in the RDO process.

[0520] Clause 106. The method of clause 104 or 105, wherein the distortion in the RDO criterion is equal to the distortion without NN filter model.

[0521] Clause 107. The method of any of clauses 85-106, wherein at least one of the first threshold, the second threshold, the third threshold, the fourth threshold, the first cost threshold, or the second cost threshold is set according to temporal layers, or wherein at least one of the first threshold, the second threshold, the third threshold, the fourth threshold, the first cost threshold, or the second cost

threshold is set according to the QP, or wherein at least one of the first threshold, the second threshold, the third threshold, the fourth threshold, the first cost threshold, or the second cost threshold is set according to a slice type, or wherein at least one of the first threshold, the second threshold, the third threshold, the fourth threshold, the first cost threshold, or the second cost threshold is set according to a configuration of the video unit.

[0522] Clause 108. The method of any of clauses 1-3, further comprising: determining whether to and/or the approach to apply the at least one NN filter in the RDO process based on a rate distortion cost without NN filter model and a rate distortion cost with NN filter.

[0523] Clause 109. The method of clause 108, wherein whether to apply the at least one NN filter model for a first coding mode is dependent on a rate distortion cost of the first coding mode without NN filter model and a rate distortion cost of a second coding mode with NN filter model.

[0524] Clause 110. The method of clause 108, wherein a rate distortion cost of the first coding mode without NN filter model is equal to the RDO criterion in the RDO process.

[0525] Clause 111. The method of clause 110, wherein the distortion in the RDO criterion is equal to the distortion without NN filter model.

[0526] Clause 112. The method of clause 108, wherein a rate distortion cost of a second coding mode with NN filter model is equal to the RDO criterion in the RDO process.

[0527] Clause 113. The method of clause 112, wherein the distortion in the RDO criterion is equal to a minimal value of the distortion without NN filter model and the distortion with the n-th NN filter model, or wherein the distortion in the RDO criterion is equal to the distortion with the n-th NN filter model.

[0528] Clause 114. The method of clause 108, wherein if a rate distortion cost of the first coding mode without NN filter model is higher than or equal to a rate distortion cost of a second coding mode with NN filter model multiplied a first scaling factor, the at least one NN filter is not applied for the first coding mode.

[0529] Clause 115. The method of any of clauses 108-114, wherein if a rate distortion cost of a second coding mode with NN filter model multiplied a second scaling factor is higher than or equal to a rate distortion cost of the first coding mode without NN filter model, the at least one NN filter is not applied for the first coding mode.

[0530] Clause 116. The method of any of clauses 108-114, wherein if a rate distortion cost of the first coding mode without NN filter model is higher than or equal to a rate distortion cost of a second coding mode with NN filter model multiplied a first scaling factor, or if the rate distortion cost of the second coding mode with NN filter model multiplied a second scaling factor is higher than or equal to the rate distortion cost of the first coding mode without NN filter model, the at least one NN filter is not applied for the first coding mode.

[0531] Clause 117. The method of any of clauses 108-114, wherein if a ratio between a rate distortion cost of the first coding mode without NN filter model and a rate distortion cost of a second coding mode with NN filter model is higher than or equal to a first ratio threshold, the at least one NN filter model is not applied.

[0532] Clause 118. The method of any of clauses 108-114, wherein if a ratio between a rate distortion cost of the first coding mode without NN filter model and a rate distortion

cost of a second coding mode with NN filter model is less than or equal to a second ratio threshold, the at least one NN filter model is not applied.

[0533] Clause 119. The method of any of clauses 108-114, wherein if a ratio between a rate distortion cost of the first coding mode without NN filter model and a rate distortion cost of a second coding mode with NN filter model is higher than or equal to a first ratio threshold, or if a ratio between the rate distortion cost of the first coding mode without NN filter model and the rate distortion cost of the second coding mode with NN filter model is less than or equal to a second ratio threshold, the at least one NN filter is not applied.

[0534] Clause 120. The method of any of clauses 108-119, wherein at least one of: the first scaling factor, the second scaling factor, the first ratio threshold, or the second ratio threshold is set according to temporal layers, or wherein at least one of: the first scaling factor, the second scaling factor, the first ratio threshold, or the second ratio threshold is set according to the QP, or wherein at least one of: the first scaling factor, the second scaling factor, the first ratio threshold, or the second ratio threshold is set according to a slice type, or wherein at least one of: the first scaling factor, the second scaling factor, the first ratio threshold, or the second ratio threshold is set according to a configuration of the video unit.

[0535] Clause 121. The method of any of clauses 1-3, wherein the at least one NN filter is only applied to partial samples of a block in the RDO process.

[0536] Clause 122. The method of clause 121, wherein the at least one NN filter is only applied to a center subblock in the block.

[0537] Clause 123. The method of clause 122, wherein a width of the subblock is equal to half of width of the block, and a height of the subblock is equal to half of height of the block.

[0538] Clause 124. The method of clause 122, wherein a width of the subblock is equal to three quarters of width of the block, and a height of the subblock is equal to three quarters of height of the block.

[0539] Clause 125. The method of any of clauses 1-3, further comprising: determining whether to and/or the approach to apply the at least one NN filter in the RDO process based on temporal layers.

[0540] Clause 126. The method of clause 125, wherein the at least one NN filter is applied to the temporal layer with ID which is greater than K, or wherein the at least one NN filter is applied to the temporal layer with ID which is smaller than K, wherein K is an integer number.

[0541] Clause 127. The method of clause 126, wherein K is equal to one of 0, 1, 2, 3, 4, 5, or 6.

[0542] Clause 128. The method of any of clauses 1-3, further comprising: determining whether to and/or the approach to apply the at least one NN filter in the RDO process based on coding statistics of sub coding units.

[0543] Clause 129. The method of clause 128, wherein if the at least one NN filter is applied to a portion or all of the sub coding units, the at least one NN filter is not applied to the video unit, or wherein if the at least one NN filter is applied to the portion or all of the sub coding units, the at least one NN filter is applied to the video unit.

[0544] Clause 130. The method of clause 128, wherein if at least one of: rate-distortion costs or distortions of the portion or all of the sub coding units are available, the at least one NN filter is not applied to the video unit, or wherein

if at least one of: rate-distortion costs or distortions of the portion or all of the sub coding units are available, the at least one NN filter is applied to the video unit.

[0545] Clause 131. The method of any of clauses 1-130, wherein the conversion includes encoding the video unit into the bitstream.

[0546] Clause 132. The method of any of clauses 1-130, wherein the conversion includes decoding the video unit from the bitstream.

[0547] Clause 133. An apparatus for video processing comprising a processor and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor, cause the processor to perform a method in accordance with any of clauses 1-132.

[0548] Clause 134. A non-transitory computer-readable storage medium storing instructions that cause a processor to perform a method in accordance with any of clauses 1-132.

[0549] Clause 135. A non-transitory computer-readable recording medium storing a bitstream of a video which is generated by a method performed by an apparatus for video processing, wherein the method comprises: determining whether to apply at least one neural network (NN) filter model or determine a rate distortion cost during a rate distortion optimization (RDO) process of a video unit of the video based on at least one of: a distortion without NN filter model, a distortion with n-th NN filter model, a combination of distortions of a plurality of NN filter models, or coding statistics of the video unit, and wherein n is an integer number; determining a coding mode of the video unit based on a rate distortion optimization (RDO) criterion in the RDO process; and generating the bitstream based on the coding mode.

[0550] Clause 136. A method for storing a bitstream of a video, comprising: determining whether to apply at least one neural network (NN) filter model or determine a rate distortion cost during a rate distortion optimization (RDO) process of a video unit of the video based on at least one of: a distortion without NN filter model, a distortion with n-th NN filter model, a combination of distortions of a plurality of NN filter models, or coding statistics of the video unit, and wherein n is an integer number; determining a coding mode of the video unit based on a rate distortion optimization (RDO) criterion in the RDO process; generating the bitstream based on the coding mode; and storing the bitstream in a non-transitory computer-readable recording medium.

Example Device

[0551] FIG. 18 illustrates a block diagram of a computing device 1800 in which various embodiments of the present disclosure can be implemented. The computing device 1800 may be implemented as or included in the source device 110 (or the video encoder 114 or 200) or the destination device 120 (or the video decoder 124 or 300).

[0552] It would be appreciated that the computing device 1800 shown in FIG. 18 is merely for purpose of illustration, without suggesting any limitation to the functions and scopes of the embodiments of the present disclosure in any manner.

[0553] As shown in FIG. 18, the computing device 1800 includes a general-purpose computing device 1800. The computing device 1800 may at least comprise one or more processors or processing units 1810, a memory 1820, a

storage unit **1830**, one or more communication units **1840**, one or more input devices **1850**, and one or more output devices **1860**.

[0554] In some embodiments, the computing device **1800** may be implemented as any user terminal or server terminal having the computing capability. The server terminal may be a server, a large-scale computing device or the like that is provided by a service provider. The user terminal may for example be any type of mobile terminal, fixed terminal, or portable terminal, including a mobile phone, station, unit, device, multimedia computer, multimedia tablet, Internet node, communicator, desktop computer, laptop computer, notebook computer, netbook computer, tablet computer, personal communication system (PCS) device, personal navigation device, personal digital assistant (PDA), audio/video player, digital camera/video camera, positioning device, television receiver, radio broadcast receiver, E-book device, gaming device, or any combination thereof, including the accessories and peripherals of these devices, or any combination thereof. It would be contemplated that the computing device **1800** can support any type of interface to a user (such as “wearable” circuitry and the like).

[0555] The processing unit **1810** may be a physical or virtual processor and can implement various processes based on programs stored in the memory **1820**. In a multi-processor system, multiple processing units execute computer executable instructions in parallel so as to improve the parallel processing capability of the computing device **1800**. The processing unit **1810** may also be referred to as a central processing unit (CPU), a microprocessor, a controller or a microcontroller.

[0556] The computing device **1800** typically includes various computer storage medium. Such medium can be any medium accessible by the computing device **1800**, including, but not limited to, volatile and non-volatile medium, or detachable and non-detachable medium. The memory **1820** can be a volatile memory (for example, a register, cache, Random Access Memory (RAM)), a non-volatile memory (such as a Read-Only Memory (ROM), Electrically Erasable Programmable Read-Only Memory (EEPROM), or a flash memory), or any combination thereof. The storage unit **1830** may be any detachable or non-detachable medium and may include a machine-readable medium such as a memory, flash memory drive, magnetic disk or another other media, which can be used for storing information and/or data and can be accessed in the computing device **1800**.

[0557] The computing device **1800** may further include additional detachable/non-detachable, volatile/non-volatile memory medium. Although not shown in FIG. **18**, it is possible to provide a magnetic disk drive for reading from and/or writing into a detachable and non-volatile magnetic disk and an optical disk drive for reading from and/or writing into a detachable non-volatile optical disk. In such cases, each drive may be connected to a bus (not shown) via one or more data medium interfaces.

[0558] The communication unit **1840** communicates with a further computing device via the communication medium. In addition, the functions of the components in the computing device **1800** can be implemented by a single computing cluster or multiple computing machines that can communicate via communication connections. Therefore, the computing device **1800** can operate in a networked environment

using a logical connection with one or more other servers, networked personal computers (PCs) or further general network nodes.

[0559] The input device **1850** may be one or more of a variety of input devices, such as a mouse, keyboard, tracking ball, voice-input device, and the like. The output device **1860** may be one or more of a variety of output devices, such as a display, loudspeaker, printer, and the like. By means of the communication unit **1840**, the computing device **1800** can further communicate with one or more external devices (not shown) such as the storage devices and display device, with one or more devices enabling the user to interact with the computing device **1800**, or any devices (such as a network card, a modem and the like) enabling the computing device **1800** to communicate with one or more other computing devices, if required. Such communication can be performed via input/output (I/O) interfaces (not shown).

[0560] In some embodiments, instead of being integrated in a single device, some or all components of the computing device **1800** may also be arranged in cloud computing architecture. In the cloud computing architecture, the components may be provided remotely and work together to implement the functionalities described in the present disclosure. In some embodiments, cloud computing provides computing, software, data access and storage service, which will not require end users to be aware of the physical locations or configurations of the systems or hardware providing these services. In various embodiments, the cloud computing provides the services via a wide area network (such as Internet) using suitable protocols. For example, a cloud computing provider provides applications over the wide area network, which can be accessed through a web browser or any other computing components. The software or components of the cloud computing architecture and corresponding data may be stored on a server at a remote position. The computing resources in the cloud computing environment may be merged or distributed at locations in a remote data center. Cloud computing infrastructures may provide the services through a shared data center, though they behave as a single access point for the users. Therefore, the cloud computing architectures may be used to provide the components and functionalities described herein from a service provider at a remote location. Alternatively, they may be provided from a conventional server or installed directly or otherwise on a client device.

[0561] The computing device **1800** may be used to implement video encoding/decoding in embodiments of the present disclosure. The memory **1820** may include one or more video coding modules **1825** having one or more program instructions. These modules are accessible and executable by the processing unit **1810** to perform the functionalities of the various embodiments described herein.

[0562] In the example embodiments of performing video encoding, the input device **1850** may receive video data as an input **1870** to be encoded. The video data may be processed, for example, by the video coding module **1825**, to generate an encoded bitstream. The encoded bitstream may be provided via the output device **1860** as an output **1880**.

[0563] In the example embodiments of performing video decoding, the input device **1850** may receive an encoded bitstream as the input **1870**. The encoded bitstream may be processed, for example, by the video coding module **1825**,

to generate decoded video data. The decoded video data may be provided via the output device **1860** as the output **1880**. **[0564]** While this disclosure has been particularly shown and described with references to preferred embodiments thereof, it will be understood by those skilled in the art that various changes in form and details may be made therein without departing from the spirit and scope of the present application as defined by the appended claims. Such variations are intended to be covered by the scope of this present application. As such, the foregoing description of embodiments of the present application is not intended to be limiting.

I/We claim:

1. A method of video processing, comprising:
 - determining, for a conversion between a video unit of a video and a bitstream of the video, whether to apply at least one neural network (NN) filter model or determine a rate distortion cost during a rate distortion optimization (RDO) process of the video unit based on at least one of: a distortion without NN filter model, a distortion with n-th NN filter model, a combination of distortions of a plurality of NN filter models, or coding statistics of the video unit, and wherein n is an integer number;
 - determining a coding mode of the video unit based on a rate distortion optimization (RDO) criterion in the RDO process; and
 - performing the conversion based on the coding mode.
2. The method of claim 1, further comprising:
 - determining an approach to apply the at least one neural network (NN) filter model or determine the rate distortion cost during the RDO process of the video unit based on at least one of: the distortion without NN filter model, the distortion with n-th NN filter model, or the combination of distortions of the plurality of NN filter models.
3. The method of claim 1, wherein the distortion with the n-th NN filter model is derived by comparing filter reconstruction samples which are filtered by the n-th NN filter model with original samples, and
 - wherein the distortion in the RDO criterion is represented as $J=D+\lambda R$, wherein D represents a distortion, λ represents a coefficient parameter, R represents a rate associated with a candidate coding mode.
4. The method of claim 1, wherein the distortion in the RDO criterion is the distortion with the n-th NN filter model, and/or
 - wherein the distortion in the RDO criterion is a minimal value of the distortion without NN filter model and the distortion with the n-th NN filter model, and/or
 - wherein the distortion in the RDO criterion is a distortion with a best NN filter model, and/or
 - wherein the distortion in the RDO criterion is a minimal value of distortion without NN filter model and a distortion with a best NN filter model, and/or
 - wherein the distortion in the RDO criterion is the distortion without NN filter model multiplied a scaling factor, and/or
 - wherein the distortion in the RDO criterion is derived according to the distortion without NN filter model and the distortion with the n-th NN filter model.
5. The method of claim 4, wherein n is one of: 1, 2, 3, or wherein the best NN filter model is selected by distortion, or

wherein the best NN filter model is a default one, or wherein the scaling factor is one of 1.0, 0.9, or 1.1.

6. The method of claim 4, wherein the distortion in the RDO criterion is derived as:

$$D = f_0 * D_{ORG} + f_1 * D_{NNLF}^n,$$

and

wherein D represents the distortion in the RDO criterion, f_0 represents a first scaling factor, f_1 represents a second scaling factor, D_{ORG} represents the distortion without NN filter model, and D_{NNLF}^n represents the distortion with the n-th NN filter model, or

wherein the distortion in the RDO criterion is derived as:

$$\min(f_0 * D_{ORG}, f_1 * D_{NNLF}^n),$$

wherein D represents the distortion in the RDO criterion, f_0 represents a first scaling factor, f_1 represents a second scaling factor, D_{ORG} represents the distortion without NN filter model, and D_{NNLF}^n represents the distortion with the n-th NN filter model.

7. The method of claim 1, wherein the distortion in the RDO criterion is a combination of one or more of:
 - the distortion with the n-th NN filter model,
 - a minimal value of the distortion without NN filter model and the distortion with the n-th NN filter model,
 - a distortion with a best NN filter model,
 - a minimal value of distortion without NN filter model and a distortion with a best NN filter model,
 - the distortion without NN filter model multiplied a scaling factor, or
 - a derivation according to the distortion without NN filter model and the distortion with the n-th NN filter model.
8. The method of claim 7, wherein the combination is dependent on coding statistics of the video unit, and/or
 - wherein if one or all of candidate modes are not partitioning mode, the distortion in the RDO criterion is a combination of at least one of: a distortion with p-th NN filter model or the distortion without NN filter model, wherein p is an integer number, and/or
 - wherein if one or all of candidate modes are partitioning modes, the distortion in the RDO criterion is a combination of at least one of: a distortion with q-th NN filter model, a distortion with p-th NN filter model, or the distortion without NN filter model, wherein p and q are integer numbers, and/or
 - wherein the combination is dependent on at least one of: usage of q-th NN filter model or usage of p-th NN filter model, wherein p and q are integer numbers, and/or
 - wherein an approach of determining the distortion in the RDO criterion is applied according to a predefined order or an adaptive order.
9. The method of claim 1, wherein the distortion in the RDO criterion is the minimal value of distortions with m NN filter models, wherein m is an integer number, or
 - wherein the distortion in the RDO criterion is a minimal value of the distortion without NN filter model and distortions with m NN filter models, wherein m is an integer number, and/or

wherein the distortion in the RDO criterion is a combination of the distortion without NN filter model and distortions with m NN filter models, wherein m is an integer number, and/or

wherein the distortion without NN filter model is dependent on another filter which is different with the at least one NN filter model.

10. The method of claim 9, wherein m is an available number of constructed NN filter models, and/or

wherein m is dependent on a type of video unit, and/or

wherein m is dependent on a signaled parameter of video unit, and/or

wherein m is a default value, and/or

wherein m is one of: 0, 1, 2, 3, or 4, and/or

wherein the distortion with the other filter is derived by comparing filter reconstruction samples which are filtered by the other filter with original samples, and/or wherein the other filter is applied on reconstruction samples before the at least one NN filter model, and/or wherein the at least one NN filter model is applied after the other filter, and/or

wherein the distortion without NN filter model is the distortion with the other filter multiplied a scaling factor, and/or

wherein the distortion without NN filter model is a combination of the distortions with a plurality of other filters.

11. The method of claim 1, wherein an input of the at least one NN filter model in the RDO process include a signal from at least one of: a current block or a neighboring block, and/or wherein the coding statistics comprises at least one of:

a prediction mode,

QP,

a temporal layer,

a slice type, or

a dimension of the video unit.

12. The method of claim 1, wherein if a width of the video unit is less than or equal to a first threshold and a height of the video unit is less than or equal to a second threshold, the at least one NN filter model is applied in the RDO process, and/or

wherein if a width of the video unit is larger than or equal to a first threshold and a height of the video unit is larger than or equal to a second threshold, the at least one NN filter model is applied in the RDO process, and/or

wherein if a value of the width of the video unit multiplying the height of the video unit is larger than or equal to a third threshold, the at least one NN filter model is applied in the RDO process, and/or

wherein if a value of the width of the video unit multiplying the height of the video unit is less than or equal to a fourth threshold, the at least one NN filter model is applied in the RDO process.

13. The method of claim 1, further comprising:

determining whether to and/or the approach to apply the at least one NN filter in the RDO process based on color components, and/or

determining whether to and/or the approach to apply the at least one NN filter in the RDO process based on a

rate distortion cost without NN filter model and a rate distortion cost with NN filter, and/or determining whether to and/or the approach to apply the at least one NN filter in the RDO process based on temporal layers.

14. The method of claim 1, wherein the at least one NN filter is only applied to partial samples of a block in the RDO process.

15. The method of claim 1, further comprising:

determining whether to and/or the approach to apply the at least one NN filter in the RDO process based on coding statistics of sub coding units.

16. The method of claim 15, wherein if the at least one NN filter is applied to a portion or all of the sub coding units, the at least one NN filter is not applied to the video unit, or

wherein if the at least one NN filter is applied to the portion or all of the sub coding units, the at least one NN filter is applied to the video unit, and/or

wherein if at least one of: rate-distortion costs or distortions of the portion or all of the sub coding units are available, the at least one NN filter is not applied to the video unit, or

wherein if at least one of: rate-distortion costs or distortions of the portion or all of the sub coding units are available, the at least one NN filter is applied to the video unit.

17. The method of claim 1, wherein the conversion includes encoding the video unit into the bitstream, or

wherein the conversion includes decoding the video unit from the bitstream.

18. An apparatus for video processing comprising a processor and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor, cause the processor to:

determine, for a conversion between a video unit of a video and a bitstream of the video, whether to apply at least one neural network (NN) filter model or determine a rate distortion cost during a rate distortion optimization (RDO) process of the video unit based on at least one of: a distortion without NN filter model, a distortion with n -th NN filter model, a combination of distortions of a plurality of NN filter models, or coding statistics of the video unit, and wherein n is an integer number;

determine a coding mode of the video unit based on a rate distortion optimization (RDO) criterion in the RDO process; and

perform the conversion based on the coding mode.

19. A non-transitory computer-readable storage medium storing instructions that cause a processor to:

determine, for a conversion between a video unit of a video and a bitstream of the video, whether to apply at least one neural network (NN) filter model or determine a rate distortion cost during a rate distortion optimization (RDO) process of the video unit based on at least one of: a distortion without NN filter model, a distortion with n -th NN filter model, a combination of distortions of a plurality of NN filter models, or coding statistics of the video unit, and wherein n is an integer number;

determine a coding mode of the video unit based on a rate distortion optimization (RDO) criterion in the RDO process; and

perform the conversion based on the coding mode.

20. A non-transitory computer-readable recording medium storing a bitstream of a video which is generated by a method performed by an apparatus for video processing, wherein the method comprises:

determining whether to apply at least one neural network (NN) filter model or determine a rate distortion cost during a rate distortion optimization (RDO) process of a video unit of the video based on at least one of: a distortion without NN filter model, a distortion with n-th NN filter model, a combination of distortions of a plurality of NN filter models, or coding statistics of the video unit, and wherein n is an integer number;

determining a coding mode of the video unit based on a rate distortion optimization (RDO) criterion in the RDO process; and

generating the bitstream based on the coding mode.

* * * * *