Innovation, Sciences et Développement économique Canada
Office de la Propriété Intellectuelle du Canada

Innovation, Science and Economic Development Canada
Canadian Intellectual Property Office

CA 3225288 A1 2021/05/27

(21) 3 225 288

(12) DEMANDE DE BREVET CANADIEN
CANADIAN PATENT APPLICATION

(13) A1

(22) Date de dépôt/Filing Date: 2020/11/20

(41) Mise à la disp. pub./Open to Public Insp.: 2021/05/27

(62) Demande originale/Original Application: 3 162 583

(30) Priorité/Priority: 2019/11/22 (US62/939,530)

(51) Cl.Int./Int.Cl. *G06T 9/00* (2006.01)

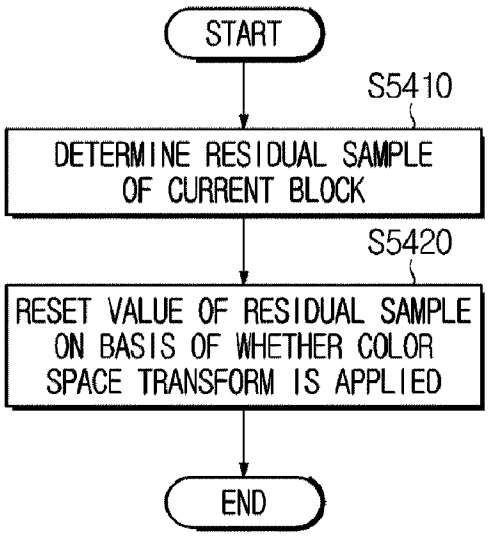
(71) Demandeur/Applicant:
LG ELECTRONICS INC., KR

(72) Inventeurs/Inventors:
ZHAO, JIE, KR;
KIM, SEUNG HWAN, KR;
SALEHIFAR, MEHDI, KR

(74) Agent: SMART & BIGGAR LP

(54) Titre : PROCEDE ET DISPOSITIF DE CODAGE/DECODAGE D'IMAGE UTILISANT UNE TRANSFORMATION DE COULEUR SANS PERTE, ET PROCEDE DE TRANSMISSION DE TRAIN DE BITS

(54) Title: IMAGE ENCODING/DECODING METHOD AND DEVICE USING LOSSLESS COLOR TRANSFORM, AND METHOD FOR TRANSMITTING BITSTREAM



(57) Abrégé/Abstract:
Provided are an image encoding/decoding method and device. The image decoding method performed by the image decoding device, according to the present disclosure, may comprise the steps of: determining a residual sample of a current block; and resetting the value of the residual sample on the basis of whether a color space transform is applied. The step for resetting the value of the residual block may be performed on the basis of a half value of a chroma residual sample value.

ABSTRACT

Provided are an image encoding/decoding method and device. The image decoding method performed by the image decoding device, according to the present disclosure, may comprise the steps of: determining a residual sample of a current block; and resetting the value of the residual sample on the basis of whether a color space transform is applied. The step for resetting the value of the residual block may be performed on the basis of a half value of a chroma residual sample value.

Date Reçue/Date Received 2023-12-29

DESCRIPTION

IMAGE ENCODING/DECODING METHOD AND DEVICE USING LOSSLESS COLOR TRANSFORM, AND METHOD FOR TRANSMITTING BITSTREAM

This is a divisional application of Canadian National Phase Application No. 3,162,583, filed on 20th November, 2020.

Technical Field

[1] The present disclosure relates to an image encoding/decoding method and device. More particularly, the present disclosure relates to an image encoding/decoding method and device using color transform, and a method for transmitting a bitstream generated by the image encoding method/device of the present disclosure.

Background Art

[2] Recently, demand for high-resolution and high-quality images such as high definition (HD) images and ultra high definition (UHD) images is increasing in various fields. As resolution and quality of image data are improved, the amount of transmitted information or bits relatively increases as compared to existing image data. An increase in the amount of transmitted information or bits causes an increase in transmission cost and storage cost.

[3] Accordingly, there is a need for high-efficient image compression technology for effectively transmitting, storing and reproducing information on high-resolution and high-quality images.

Disclosure

Technical Problem

[4] The present disclosure is directed to providing an image encoding/decoding method and device with improved encoding/decoding efficiency.

[5] In addition, the present disclosure is directed to providing an image encoding/decoding method and device for improving encoding/decoding efficiency by performing lossless color transform.

[6] In addition, the present disclosure is directed to providing a method for transmitting a bitstream generated by an image encoding method or device according to the present disclosure.

[7] In addition, the present disclosure is directed to providing a recording medium storing therein a bitstream generated by an image encoding method or device according to the present disclosure.

[8] In addition, the present disclosure is directed to providing a recording medium storing therein a bitstream that is received, decoded, and used to reconstruct an image by an image decoding device according to the present disclosure.

[9] It will be appreciated by persons skilled in the art that technical objectives to be achieved in the present disclosure are not limited to the above-mentioned technical objectives

Date Reçue/Date Received 2023-12-29

and other technical objectives which are not described herein will be clearly understood from the following description.

Technical Solution

[10] According to an aspect of the present disclosure, there is provided an image decoding method performed by an image decoding device, the method including: determining a residual sample of a current block; and resetting a value of the residual sample on the basis of whether a color space transform is applied. Herein, the resetting of the value of the residual sample may be performed on the basis of a half value of a chroma residual sample value.

[11] In addition, according to another aspect of the present disclosure, there is provided an image decoding device including a memory and at least one processor, wherein the at least one processor is configured to determine a residual sample of a current block, and reset a value of the residual sample on the basis of whether a color space transform is applied. Herein, the processor may be configured to reset the value of the residual sample on the basis of a half value of a chroma residual sample value.

[12] In addition, according to another aspect of the present disclosure, there is provided an image encoding method performed by an image encoding device, the method including: determining a residual sample of a current block; and resetting a value of the residual sample on the basis of whether a color space transform is applied. Herein, the resetting of the value of the residual sample may be performed on the basis of a half value of a chroma residual sample value.

[13] In addition, according to another aspect of the present disclosure, there is provided a transmission method for transmitting a bitstream generated by an image encoding device or an image encoding method of the present disclosure.

[14] In addition, according to another aspect of the present disclosure, there is provided a computer-readable recording medium storing therein a bitstream generated by an image encoding method or an image encoding device of the present disclosure.

[15] The features briefly summarized above with respect to the present disclosure are merely exemplary aspects of the detailed description below of the present disclosure, and do not limit the scope of the present disclosure.

Advantageous Effects

[16] According to the present disclosure, it is possible to provide an image encoding/decoding method and device with improved encoding/decoding efficiency.

[17] In addition, according to the present disclosure, it is possible to provide an image encoding/decoding method and device capable of improving encoding/decoding efficiency by performing lossless color transform.

Date Reçue/Date Received 2023-12-29

[18] In addition, according to the present disclosure, it is possible to provide a method for transmitting a bitstream generated by an image encoding method or device according to the present disclosure.

[19] In addition, according to the present disclosure, it is possible to provide a recording medium storing therein a bitstream generated by an image encoding method or device according to the present disclosure.

[20] In addition, according to the present disclosure, it is possible to provide a recording medium storing therein a bitstream that is received, decoded, and used to reconstruct an image by an image decoding device according to the present disclosure.

[21] It will be appreciated by persons skilled in the art that the effects that can be achieved through the present disclosure are not limited to what has been particularly described hereinabove and other advantages of the present disclosure will be more clearly understood from the following description.

Description of Drawings

[22] FIG. 1 is a view schematically showing a video coding system, to which an embodiment of the present disclosure is applicable.

[23] FIG. 2 is a view schematically showing an image encoding device, to which an embodiment of the present disclosure is applicable.

[24] FIG. 3 is a view schematically showing an image decoding device, to which an embodiment of the present disclosure is applicable.

[25] FIG. 4 is a view showing a partitioning structure of an image according to an embodiment.

[26] FIG. 5 is a view showing an embodiment of a partitioning type of a block according to a multi-type tree structure.

[27] FIG. 6 is a view showing a signaling mechanism of block splitting information in a structure of a quadtree with a nested multi-type tree according to the present disclosure.

[28] FIG. 7 is a view showing an embodiment in which a CTU is partitioned into multiple CUs.

[29] FIG. 8 is a view showing neighboring reference samples according to an embodiment.

[30] FIGS. 9 and 10 are views showing intra prediction according to an embodiment.

[31] FIG. 11 is a view showing an encoding method using inter prediction according to an embodiment.

[32] FIG. 12 is a view showing a decoding method using inter prediction according to an embodiment.

[33] FIG. 13 is a view showing a block diagram of CABAC according to an embodiment for encoding one syntax element.

Date Reçue/Date Received 2023-12-29

[34] FIGS. 14 to 17 are views showing entropy encoding and decoding according to an embodiment.

[35] FIGS. 18 and 19 are views showing examples of picture decoding and encoding procedures according to an embodiment.

[36] FIG. 20 is a view showing a hierarchical structure for a coded image according to an embodiment.

[37] FIG. 21 is a view showing an embodiment of a decoding process to which ACT is applied.

[38] FIG. 22 is a view showing an embodiment of a sequence parameter set syntax table in which a syntax element related to ACT is signaled.

[39] FIGS. 23 to 29 are views successively showing an embodiment of a syntax table of an encoding basis in which a syntax element related to ACT is signaled.

[40] FIG. 30 is a view showing coding tree syntax according to an embodiment.

[41] FIG. 31 is a view showing a method of encoding a residual sample of BDPCM according to an embodiment.

[42] FIG. 32 is a view showing modified quantized residual blocks generated by performing BDPCM according to an embodiment.

[43] FIG. 33 is a flowchart showing a procedure for encoding a current block by applying BDPCM in an image encoding device according to an embodiment.

[44] FIG. 34 is a flowchart showing a procedure for reconstructing a current block by applying BDPCM in an image decoding device according to an embodiment.

[45] FIGS. 35 to 37 are views schematically showing syntax for signaling information on BDPCM.

[46] FIGS. 38 to 53 are views showing a syntax table for signaling an ACT syntax element according to each individual embodiment of the present disclosure.

[47] FIGS. 54 and 55 are views showing an image decoding method according to an embodiment.

[48] FIGS. 56 and 57 are views showing an image encoding method according to an embodiment.

[49] FIG. 58 is a view showing a content streaming system, to which an embodiment of the present disclosure is applicable.

Mode for Invention

[50] Hereinafter, the embodiments of the present disclosure will be described in detail with reference to the accompanying drawings so as to be easily implemented by those skilled in the art. However, the present disclosure may be implemented in various different forms, and is not limited to the embodiments described herein.

Date Reçue/Date Received 2023-12-29

[51] In describing the present disclosure, if it is determined that the detailed description of a related known function or construction renders the scope of the present disclosure unnecessarily ambiguous, the detailed description thereof will be omitted. In the drawings, parts not related to the description of the present disclosure are omitted, and similar reference numerals are attached to similar parts.

[52] In the present disclosure, when a component is “connected”, “coupled” or “linked” to another component, it may include not only a direct connection relationship but also an indirect connection relationship in which an intervening component is present. In addition, when a component “includes” or “has” other components, it means that other components may be further included, rather than excluding other components unless otherwise stated.

[53] In the present disclosure, the terms first, second, etc. may be used only for the purpose of distinguishing one component from other components, and do not limit the order or importance of the components unless otherwise stated. Accordingly, within the scope of the present disclosure, a first component in one embodiment may be referred to as a second component in another embodiment, and similarly, a second component in one embodiment may be referred to as a first component in another embodiment.

[54] In the present disclosure, components that are distinguished from each other are intended to clearly describe each feature, and do not mean that the components are necessarily separated. That is, a plurality of components may be integrated and implemented in one hardware or software unit, or one component may be distributed and implemented in a plurality of hardware or software units. Therefore, even if not stated otherwise, such embodiments in which the components are integrated or the component is distributed are also included in the scope of the present disclosure.

[55] In the present disclosure, the components described in various embodiments do not necessarily mean essential components, and some components may be optional components. Accordingly, an embodiment consisting of a subset of components described in an embodiment is also included in the scope of the present disclosure. In addition, embodiments including other components in addition to components described in the various embodiments are included in the scope of the present disclosure.

[56] The present disclosure relates to encoding and decoding of an image, and terms used in the present disclosure may have a general meaning commonly used in the technical field, to which the present disclosure belongs, unless newly defined in the present disclosure.

[57] In the present disclosure, a "video" may mean a set of images in a series according to the passage of time. A "picture" generally means the basis representing one image in a particular time period, and a slice/tile is an encoding basis constituting a part of a picture in encoding. One picture may be composed of one or more slices/tiles. In addition, a slice/tile may include one or more coding tree units (CTUs). One picture may be composed of one or more slices/tiles. One picture may be composed of one or more tile groups. One tile group may

Date Reçue/Date Received 2023-12-29

include one or more tiles. A brick may refer to a quadrangular area of CTU rows within a tile in a picture. One tile may include one or more bricks. A brick may refer to a quadrangular area of CTU rows within a tile. One tile may be partitioned into a plurality of bricks, and each brick may include one or more CTU rows belonging to a tile. A tile that is not partitioned into a plurality of bricks may also be treated as a brick.

[58] In the present disclosure, a "pixel" or a "pel" may mean a smallest unit constituting one picture (or image). In addition, "sample" may be used as a term corresponding to a pixel. A sample may generally represent a pixel or a value of a pixel, and may represent only a pixel/pixel value of a luma component or only a pixel/pixel value of a chroma component.

[59] In the present disclosure, a "unit" may represent a basic unit of image processing. The unit may include at least one of a specific region of the picture and information related to the region. One unit may include one luma block and two chroma (e.g., Cb, Cr) blocks. The unit may be used interchangeably with terms such as "sample array", "block" or "area" in some cases. In a general case, an M×N block may include samples (or sample arrays) or a set (or array) of transform coefficients of M columns and N rows.

[60] In the present disclosure, "current block" may mean one of "current coding block", "current coding unit", "coding target block", "decoding target block" or "processing target block". When prediction is performed, "current block" may mean "current prediction block" or "prediction target block". When transform (inverse transform)/quantization (dequantization) is performed, "current block" may mean "current transform block" or "transform target block". When filtering is performed, "current block" may mean "filtering target block".

[61] In addition, in the present disclosure, a "current block" may mean "a luma block of a current block" unless explicitly stated as a chroma block. The "chroma block of the current block" may be expressed by including an explicit description of a chroma block, such as "chroma block" or "current chroma block".

[62] In the present disclosure, the term "/" and "," should be interpreted to indicate "and/or". For instance, the expression "A/B" and "A, B" may mean "A and/or B." Further, "A/B/C" and "A, B, C" may mean "at least one of A, B, and/or C."

[63] In the present disclosure, the term "or" should be interpreted to indicate "and/or." For instance, the expression "A or B" may comprise 1) only "A", 2) only "B", and/or 3) both "A and B". In other words, in the present disclosure, the term "or" should be interpreted to indicate "additionally or alternatively."

[64] <u>Overview of video coding system</u>

[65] FIG. 1 is a view showing a video coding system according to the present disclosure.

[66] The video coding system according to an embodiment may include a source device 10 and a receiving device 20. The source device 10 may deliver encoded video and/or image information or data to the receiving device 20 in the form of a file or streaming via a digital storage medium or network.

Date Reçue/Date Received 2023-12-29

[67] The source device 10 according to an embodiment may include a video source generator 11, an encoding apparatus 12 and a transmitter 13. The receiving device 20 according to an embodiment may include a receiver 21, a decoding apparatus 12 and a renderer 23. The encoding apparatus 12 may be called a video/image encoding apparatus, and the decoding apparatus 12 may be called a video/image decoding apparatus. The transmitter 13 may be included in the encoding apparatus 12. The receiver 21 may be included in the decoding apparatus 12. The renderer 23 may include a display and the display may be configured as a separate device or an external component.

[68] The video source generator 11 may acquire a video/image through a process of capturing, synthesizing or generating the video/image. The video source generator 11 may include a video/image capture device and/or a video/image generating device. The video/image capture device may include, for example, one or more cameras, video/image archives including previously captured video/images, and the like. The video/image generating device may include, for example, computers, tablets and smartphones, and may (electronically) generate video/images. For example, a virtual video/image may be generated through a computer or the like. In this case, the video/image capturing process may be replaced by a process of generating related data.

[69] The encoding apparatus 12 may encode an input video/image. The encoding apparatus 12 may perform a series of procedures such as prediction, transform, and quantization for compression and coding efficiency. The encoding apparatus 12 may output encoded data (encoded video/image information) in the form of a bitstream.

[70] The transmitter 13 may transmit the encoded video/image information or data output in the form of a bitstream to the receiver 21 of the receiving device 20 through a digital storage medium or a network in the form of a file or streaming. The digital storage medium may include various storage mediums such as USB, SD, CD, DVD, Blu-ray, HDD, SSD, and the like. The transmitter 13 may include an element for generating a media file through a predetermined file format and may include an element for transmission through a broadcast/communication network. The receiver 21 may extract/receive the bitstream from the storage medium or network and transmit the bitstream to the decoding apparatus 12.

[71] The decoding apparatus 12 may decode the video/image by performing a series of procedures such as dequantization, inverse transform, and prediction corresponding to the operation of the encoding apparatus 12.

[72] The renderer 23 may render the decoded video/image. The rendered video/image may be displayed through the display.

[73] <u>Overview of image encoding apparatus</u>

[74] FIG. 2 is a view schematically showing an image encoding apparatus, to which an embodiment of the present disclosure is applicable.

Date Reçue/Date Received 2023-12-29

[75] As shown in FIG. 2, the image encoding apparatus 100 may include an image partitioner 110, a subtractor 115, a transformer 120, a quantizer 130, a dequantizer 140, an inverse transformer 150, an adder 155, a filter 160, a memory 170, an inter prediction unit 180, an intra prediction unit 185 and an entropy encoder 190. The inter prediction unit 180 and the intra prediction unit 185 may be collectively referred to as a "prediction unit". The transformer 120, the quantizer 130, the dequantizer 140 and the inverse transformer 150 may be included in a residual processor. The residual processor may further include the subtractor 115.

[76] All or at least some of the plurality of components configuring the image encoding apparatus 100 may be configured by one hardware component (e.g., an encoder or a processor) in some embodiments. In addition, the memory 170 may include a decoded picture buffer (DPB) and may be configured by a digital storage medium.

[77] The image partitioner 110 may partition an input image (or a picture or a frame) input to the image encoding apparatus 100 into one or more processing units. For example, the processing unit may be called a coding unit (CU). The coding unit may be acquired by recursively partitioning a coding tree unit (CTU) or a largest coding unit (LCU) according to a quad-tree binary-tree ternary-tree (QT/BT/TT) structure. For example, one coding unit may be partitioned into a plurality of coding units of a deeper depth based on a quad tree structure, a binary tree structure, and/or a ternary structure. For partitioning of the coding unit, a quad tree structure may be applied first and the binary tree structure and/or ternary structure may be applied later. The coding procedure according to the present disclosure may be performed based on the final coding unit that is no longer partitioned. The largest coding unit may be used as the final coding unit or the coding unit of deeper depth acquired by partitioning the largest coding unit may be used as the final coding unit. Here, the coding procedure may include a procedure of prediction, transform, and reconstruction, which will be described later. As another example, the processing unit of the coding procedure may be a prediction unit (PU) or a transform unit (TU). The prediction unit and the transform unit may be split or partitioned from the final coding unit. The prediction unit may be a unit of sample prediction, and the transform unit may be a unit for deriving a transform coefficient and/or a unit for deriving a residual signal from the transform coefficient.

[78] The prediction unit (the inter prediction unit 180 or the intra prediction unit 185) may perform prediction on a block to be processed (current block) and generate a predicted block including prediction samples for the current block. The prediction unit may determine whether intra prediction or inter prediction is applied on a current block or CU basis. The prediction unit may generate various information related to prediction of the current block and transmit the generated information to the entropy encoder 190. The information on the prediction may be encoded in the entropy encoder 190 and output in the form of a bitstream.

[79] The intra prediction unit 185 may predict the current block by referring to the samples in the current picture. The referred samples may be located in the neighborhood of the current

Date Reçue/Date Received 2023-12-29

block or may be located apart according to the intra prediction mode and/or the intra prediction technique. The intra prediction modes may include a plurality of non-directional modes and a plurality of directional modes. The non-directional mode may include, for example, a DC mode and a planar mode. The directional mode may include, for example, 33 directional prediction modes or 65 directional prediction modes according to the degree of detail of the prediction direction. However, this is merely an example, more or less directional prediction modes may be used depending on a setting. The intra prediction unit 185 may determine the prediction mode applied to the current block by using a prediction mode applied to a neighboring block.

[80] The inter prediction unit 180 may derive a predicted block for the current block based on a reference block (reference sample array) specified by a motion vector on a reference picture. In this case, in order to reduce the amount of motion information transmitted in the inter prediction mode, the motion information may be predicted in units of blocks, subblocks, or samples based on correlation of motion information between the neighboring block and the current block. The motion information may include a motion vector and a reference picture index. The motion information may further include inter prediction direction (L0 prediction, L1 prediction, Bi prediction, etc.) information. In the case of inter prediction, the neighboring block may include a spatial neighboring block present in the current picture and a temporal neighboring block present in the reference picture. The reference picture including the reference block and the reference picture including the temporal neighboring block may be the same or different. The temporal neighboring block may be called a collocated reference block, a co-located CU (colCU), and the like. The reference picture including the temporal neighboring block may be called a collocated picture (colPic). For example, the inter prediction unit 180 may configure a motion information candidate list based on neighboring blocks and generate information indicating which candidate is used to derive a motion vector and/or a reference picture index of the current block. Inter prediction may be performed based on various prediction modes. For example, in the case of a skip mode and a merge mode, the inter prediction unit 180 may use motion information of the neighboring block as motion information of the current block. In the case of the skip mode, unlike the merge mode, the residual signal may not be transmitted. In the case of the motion vector prediction (MVP) mode, the motion vector of the neighboring block may be used as a motion vector predictor, and the motion vector of the current block may be signaled by encoding a motion vector difference and an indicator for a motion vector predictor. The motion vector difference may mean a difference between the motion vector of the current block and the motion vector predictor.

[81] The prediction unit may generate a prediction signal based on various prediction methods and prediction techniques described below. For example, the prediction unit may not only apply intra prediction or inter prediction but also simultaneously apply both intra prediction and inter prediction, in order to predict the current block. A prediction method of simultaneously applying both intra prediction and inter prediction for prediction of the current

Date Reçue/Date Received 2023-12-29

block may be called combined inter and intra prediction (CIIP). In addition, the prediction unit may perform intra block copy (IBC) for prediction of the current block. Intra block copy may be used for content image/video coding of a game or the like, for example, screen content coding (SCC). IBC is a method of predicting a current picture using a previously reconstructed reference block in the current picture at a location apart from the current block by a predetermined distance. When IBC is applied, the location of the reference block in the current picture may be encoded as a vector (block vector) corresponding to the predetermined distance. IBC basically performs prediction in the current picture, but may be performed similarly to inter prediction in that a reference block is derived within the current picture. That is, IBC may use at least one of the inter prediction techniques described in the present disclosure.

[82] The prediction signal generated by the prediction unit may be used to generate a reconstructed signal or to generate a residual signal. The subtractor 115 may generate a residual signal (residual block or residual sample array) by subtracting the prediction signal (predicted block or prediction sample array) output from the prediction unit from the input image signal (original block or original sample array). The generated residual signal may be transmitted to the transformer 120.

[83] The transformer 120 may generate transform coefficients by applying a transform technique to the residual signal. For example, the transform technique may include at least one of a discrete cosine transform (DCT), a discrete sine transform (DST), a karhunen-loève transform (KLT), a graph-based transform (GBT), or a conditionally non-linear transform (CNT). Here, the GBT means transform obtained from a graph when relationship information between pixels is represented by the graph. The CNT refers to transform acquired based on a prediction signal generated using all previously reconstructed pixels. In addition, the transform process may be applied to square pixel blocks having the same size or may be applied to blocks having a variable size rather than square.

[84] The quantizer 130 may quantize the transform coefficients and transmit them to the entropy encoder 190. The entropy encoder 190 may encode the quantized signal (information on the quantized transform coefficients) and output a bitstream. The information on the quantized transform coefficients may be referred to as residual information. The quantizer 130 may rearrange quantized transform coefficients in a block form into a one-dimensional vector form based on a coefficient scanning order and generate information on the quantized transform coefficients based on the quantized transform coefficients in the one-dimensional vector form.

[85] The entropy encoder 190 may perform various encoding methods such as, for example, exponential Golomb, context-adaptive variable length coding (CAVLC), context-adaptive binary arithmetic coding (CABAC), and the like. The entropy encoder 190 may encode information necessary for video/image reconstruction other than quantized transform coefficients (e.g., values of syntax elements, etc.) together or separately. Encoded information (e.g., encoded video/image information) may be transmitted or stored in units of network

Date Reçue/Date Received 2023-12-29

abstraction layers (NALs) in the form of a bitstream. The video/image information may further include information on various parameter sets such as an adaptation parameter set (APS), a picture parameter set (PPS), a sequence parameter set (SPS), or a video parameter set (VPS). In addition, the video/image information may further include general constraint information. The signaled information, transmitted information and/or syntax elements described in the present disclosure may be encoded through the above-described encoding procedure and included in the bitstream.

[86] The bitstream may be transmitted over a network or may be stored in a digital storage medium. The network may include a broadcasting network and/or a communication network, and the digital storage medium may include various storage media such as USB, SD, CD, DVD, Blu-ray, HDD, SSD, and the like. A transmitter (not shown) transmitting a signal output from the entropy encoder 190 and/or a storage unit (not shown) storing the signal may be included as internal/external element of the image encoding apparatus 100. Alternatively, the transmitter may be provided as the component of the entropy encoder 190.

[87] The quantized transform coefficients output from the quantizer 130 may be used to generate a residual signal. For example, the residual signal (residual block or residual samples) may be reconstructed by applying dequantization and inverse transform to the quantized transform coefficients through the dequantizer 140 and the inverse transformer 150.

[88] The adder 155 adds the reconstructed residual signal to the prediction signal output from the inter prediction unit 180 or the intra prediction unit 185 to generate a reconstructed signal (reconstructed picture, reconstructed block, reconstructed sample array). If there is no residual for the block to be processed, such as a case where the skip mode is applied, the predicted block may be used as the reconstructed block. The adder 155 may be called a reconstructor or a reconstructed block generator. The generated reconstructed signal may be used for intra prediction of a next block to be processed in the current picture and may be used for inter prediction of a next picture through filtering as described below.

[89] The filter 160 may improve subjective/objective image quality by applying filtering to the reconstructed signal. For example, the filter 160 may generate a modified reconstructed picture by applying various filtering methods to the reconstructed picture and store the modified reconstructed picture in the memory 170, specifically, a DPB of the memory 170. The various filtering methods may include, for example, deblocking filtering, a sample adaptive offset, an adaptive loop filter, a bilateral filter, and the like. The filter 160 may generate various information related to filtering and transmit the generated information to the entropy encoder 190 as described later in the description of each filtering method. The information related to filtering may be encoded by the entropy encoder 190 and output in the form of a bitstream.

[90] The modified reconstructed picture transmitted to the memory 170 may be used as the reference picture in the inter prediction unit 180. When inter prediction is applied through the

Date Reçue/Date Received 2023-12-29

image encoding apparatus 100, prediction mismatch between the image encoding apparatus 100 and the image decoding apparatus may be avoided and encoding efficiency may be improved.

[91] The DPB of the memory 170 may store the modified reconstructed picture for use as a reference picture in the inter prediction unit 180. The memory 170 may store the motion information of the block from which the motion information in the current picture is derived (or encoded) and/or the motion information of the blocks in the picture that have already been reconstructed. The stored motion information may be transmitted to the inter prediction unit 180 and used as the motion information of the spatial neighboring block or the motion information of the temporal neighboring block. The memory 170 may store reconstructed samples of reconstructed blocks in the current picture and may transfer the reconstructed samples to the intra prediction unit 185.

[92] **Overview of image decoding apparatus**

[93] FIG. 3 is a view schematically showing an image decoding apparatus, to which an embodiment of the present disclosure is applicable.

[94] As shown in FIG. 3, the image decoding apparatus 200 may include an entropy decoder 210, a dequantizer 220, an inverse transformer 230, an adder 235, a filter 240, a memory 250, an inter prediction unit 260 and an intra prediction unit 265. The inter prediction unit 260 and the intra prediction unit 265 may be collectively referred to as a "prediction unit". The dequantizer 220 and the inverse transformer 230 may be included in a residual processor.

[95] All or at least some of a plurality of components configuring the image decoding apparatus 200 may be configured by a hardware component (e.g., a decoder or a processor) according to an embodiment. In addition, the memory 250 may include a decoded picture buffer (DPB) or may be configured by a digital storage medium.

[96] The image decoding apparatus 200, which has received a bitstream including video/image information, may reconstruct an image by performing a process corresponding to a process performed by the image encoding apparatus 100 of FIG. 2. For example, the image decoding apparatus 200 may perform decoding using a processing unit applied in the image encoding apparatus. Thus, the processing unit of decoding may be a coding unit, for example. The coding unit may be acquired by partitioning a coding tree unit or a largest coding unit. The reconstructed image signal decoded and output through the image decoding apparatus 200 may be reproduced through a reproducing apparatus (not shown).

[97] The image decoding apparatus 200 may receive a signal output from the image encoding apparatus of FIG. 2 in the form of a bitstream. The received signal may be decoded through the entropy decoder 210. For example, the entropy decoder 210 may parse the bitstream to derive information (e.g., video/image information) necessary for image reconstruction (or picture reconstruction). The video/image information may further include information on various parameter sets such as an adaptation parameter set (APS), a picture

Date Reçue/Date Received 2023-12-29

parameter set (PPS), a sequence parameter set (SPS), or a video parameter set (VPS). In addition, the video/image information may further include general constraint information. The image decoding apparatus may further decode picture based on the information on the parameter set and/or the general constraint information. Signaled/received information and/or syntax elements described in the present disclosure may be decoded through the decoding procedure and obtained from the bitstream. For example, the entropy decoder 210 decodes the information in the bitstream based on a coding method such as exponential Golomb coding, CAVLC, or CABAC, and output values of syntax elements required for image reconstruction and quantized values of transform coefficients for residual. More specifically, the CABAC entropy decoding method may receive a bin corresponding to each syntax element in the bitstream, determine a context model using a decoding target syntax element information, decoding information of a neighboring block and a decoding target block or information of a symbol/bin decoded in a previous stage, and perform arithmetic decoding on the bin by predicting a probability of occurrence of a bin according to the determined context model, and generate a symbol corresponding to the value of each syntax element. In this case, the CABAC entropy decoding method may update the context model by using the information of the decoded symbol/bin for a context model of a next symbol/bin after determining the context model. The information related to the prediction among the information decoded by the entropy decoder 210 may be provided to the prediction unit (the inter prediction unit 260 and the intra prediction unit 265), and the residual value on which the entropy decoding was performed in the entropy decoder 210, that is, the quantized transform coefficients and related parameter information, may be input to the dequantizer 220. In addition, information on filtering among information decoded by the entropy decoder 210 may be provided to the filter 240. Meanwhile, a receiver (not shown) for receiving a signal output from the image encoding apparatus may be further configured as an internal/external element of the image decoding apparatus 200, or the receiver may be a component of the entropy decoder 210.

[98] Meanwhile, the image decoding apparatus according to the present disclosure may be referred to as a video/image/picture decoding apparatus. The image decoding apparatus may be classified into an information decoder (video/image/picture information decoder) and a sample decoder (video/image/picture sample decoder). The information decoder may include the entropy decoder 210. The sample decoder may include at least one of the dequantizer 220, the inverse transformer 230, the adder 235, the filter 240, the memory 250, the inter prediction unit 260 or the intra prediction unit 265.

[99] The dequantizer 220 may dequantize the quantized transform coefficients and output the transform coefficients. The dequantizer 220 may rearrange the quantized transform coefficients in the form of a two-dimensional block. In this case, the rearrangement may be performed based on the coefficient scanning order performed in the image encoding apparatus. The dequantizer 220 may perform dequantization on the quantized transform coefficients by

Date Reçue/Date Received 2023-12-29

using a quantization parameter (e.g., quantization step size information) and obtain transform coefficients.

[100] The inverse transformer 230 may inversely transform the transform coefficients to obtain a residual signal (residual block, residual sample array).

[101] The prediction unit may perform prediction on the current block and generate a predicted block including prediction samples for the current block. The prediction unit may determine whether intra prediction or inter prediction is applied to the current block based on the information on the prediction output from the entropy decoder 210 and may determine a specific intra/inter prediction mode (prediction technique).

[102] It is the same as described in the prediction unit of the image encoding apparatus 100 that the prediction unit may generate the prediction signal based on various prediction methods (techniques) which will be described later.

[103] The intra prediction unit 265 may predict the current block by referring to the samples in the current picture. The description of the intra prediction unit 185 is equally applied to the intra prediction unit 265.

[104] The inter prediction unit 260 may derive a predicted block for the current block based on a reference block (reference sample array) specified by a motion vector on a reference picture. In this case, in order to reduce the amount of motion information transmitted in the inter prediction mode, the motion information may be predicted in units of blocks, subblocks, or samples based on correlation of motion information between the neighboring block and the current block. The motion information may include a motion vector and a reference picture index. The motion information may further include inter prediction direction (L0 prediction, L1 prediction, Bi prediction, etc.) information. In the case of inter prediction, the neighboring block may include a spatial neighboring block present in the current picture and a temporal neighboring block present in the reference picture. For example, the inter prediction unit 260 may configure a motion information candidate list based on neighboring blocks and derive a motion vector of the current block and/or a reference picture index based on the received candidate selection information. Inter prediction may be performed based on various prediction modes, and the information on the prediction may include information indicating a mode of inter prediction for the current block.

[105] The adder 235 may generate a reconstructed signal (reconstructed picture, reconstructed block, reconstructed sample array) by adding the obtained residual signal to the prediction signal (predicted block, predicted sample array) output from the prediction unit (including the inter prediction unit 260 and/or the intra prediction unit 265). If there is no residual for the block to be processed, such as a case where the skip mode is applied, the predicted block may be used as the reconstructed block. The description of the adder 155 is equally applicable to the adder 235. The adder 235 may be called a reconstructor or a reconstructed block generator. The generated reconstructed signal may be used for intra

Date Reçue/Date Received 2023-12-29

prediction of a next block to be processed in the current picture and may be used for inter prediction of a next picture through filtering as described below.

[106] The filter 240 may improve subjective/objective image quality by applying filtering to the reconstructed signal. For example, the filter 240 may generate a modified reconstructed picture by applying various filtering methods to the reconstructed picture and store the modified reconstructed picture in the memory 250, specifically, a DPB of the memory 250. The various filtering methods may include, for example, deblocking filtering, a sample adaptive offset, an adaptive loop filter, a bilateral filter, and the like.

[107] The (modified) reconstructed picture stored in the DPB of the memory 250 may be used as a reference picture in the inter prediction unit 260. The memory 250 may store the motion information of the block from which the motion information in the current picture is derived (or decoded) and/or the motion information of the blocks in the picture that have already been reconstructed. The stored motion information may be transmitted to the inter prediction unit 260 so as to be utilized as the motion information of the spatial neighboring block or the motion information of the temporal neighboring block. The memory 250 may store reconstructed samples of reconstructed blocks in the current picture and transfer the reconstructed samples to the intra prediction unit 265.

[108] In the present disclosure, the embodiments described in the filter 160, the inter prediction unit 180, and the intra prediction unit 185 of the image encoding apparatus 100 may be equally or correspondingly applied to the filter 240, the inter prediction unit 260, and the intra prediction unit 265 of the image decoding apparatus 200.

[109] <u>Overview of image partitioning</u>

[110] The video/image coding method according to the present disclosure may be performed based on an image partitioning structure as follows. Specifically, the procedures of prediction, residual processing ((inverse) transform, (de)quantization, etc.), syntax element coding, and filtering, which will be described later, may be performed based on a CTU, CU (and/or TU, PU) derived based on the image partitioning structure. The image may be partitioned in block units and the block partitioning procedure may be performed in the image partitioner 110 of the encoding apparatus. The partitioning related information may be encoded by the entropy encoder 190 and transmitted to the image decoding apparatus in the form of a bitstream. The entropy decoder 210 of the image decoding apparatus may derive a block partitioning structure of the current picture based on the partitioning related information obtained from the bitstream, and based on this, may perform a series of procedures (e.g., prediction, residual processing, block/picture reconstruction, in-loop filtering, etc.) for image decoding.

[111] Pictures may be partitioned into a sequence of coding tree units (CTUs). FIG. 4 shows an example in which a picture is partitioned into CTUs. The CTU may correspond to a coding tree block (CTB). Alternatively, the CTU may include a coding tree block of luma samples and two coding tree blocks of corresponding chroma samples. For example, for a picture that

Date Reçue/Date Received 2023-12-29

contains three sample arrays, the CTU may include an N×N block of luma samples and two corresponding blocks of chroma samples.

[112] Overview of partitioning of CTU

[113] As described above, the coding unit may be acquired by recursively partitioning the coding tree unit (CTU) or the largest coding unit (LCU) according to a quad-tree/binary-tree/ternary-tree (QT/BT/TT) structure. For example, the CTU may be first partitioned into quadtree structures. Thereafter, leaf nodes of the quadtree structure may be further partitioned by a multi-type tree structure.

[114] Partitioning according to quadtree means that a current CU (or CTU) is partitioned into equally four. By partitioning according to quadtree, the current CU may be partitioned into four CUs having the same width and the same height. When the current CU is no longer partitioned into the quadtree structure, the current CU corresponds to the leaf node of the quad-tree structure. The CU corresponding to the leaf node of the quadtree structure may be no longer partitioned and may be used as the above-described final coding unit. Alternatively, the CU corresponding to the leaf node of the quadtree structure may be further partitioned by a multi-type tree structure.

[115] FIG. 5 is a view showing a partitioning type of a block according to a multi-type tree structure. Splitting according to the multi-type tree structure may include two types of splitting according to a binary tree structure and two types of splitting according to a ternary tree structure.

[116] The two types of splitting according to the binary tree structure may include vertical binary splitting (SPLIT_BT_VER) and horizontal binary splitting (SPLIT_BT_HOR). Vertical binary splitting (SPLIT_BT_VER) means that the current CU is split into equally two in the vertical direction. As shown in FIG. 4, by vertical binary splitting, two CUs having the same height as the current CU and having a width which is half the width of the current CU may be generated. Horizontal binary splitting (SPLIT_BT_HOR) means that the current CU is split into equally two in the horizontal direction. As shown in FIG. 5, by horizontal binary splitting, two CUs having a height which is half the height of the current CU and having the same width as the current CU may be generated.

[117] Two types of splitting according to the ternary tree structure may include vertical ternary splitting (SPLIT_TT_VER) and horizontal ternary splitting (SPLIT_TT_HOR). In vertical ternary splitting (SPLIT_TT_VER), the current CU is split in the vertical direction at a ratio of 1:2:1. As shown in FIG. 5, by vertical ternary splitting, two CUs having the same height as the current CU and having a width which is 1/4 of the width of the current CU and a CU having the same height as the current CU and having a width which is half the width of the current CU may be generated. In horizontal ternary splitting (SPLIT_TT_HOR), the current CU is split in the horizontal direction at a ratio of 1:2:1. As shown in FIG. 4, by horizontal ternary splitting, two CUs having a height which is 1/4 of the height of the current CU and

Date Reçue/Date Received 2023-12-29

having the same width as the current CU and a CU having a height which is half the height of the current CU and having the same width as the current CU may be generated.

[118] FIG. 6 is a view showing a signaling mechanism of block splitting information in a structure of a quadtree with a nested multi-type tree according to the present disclosure.

[119] Here, the CTU is treated as the root node of the quadtree, and is partitioned for the first time into a quadtree structure. Information (e.g., qt_split_flag) indicating whether quadtree splitting is performed with respect to the current CU (CTU or node (QT_node) of the quadtree) is signaled. For example, when the qt_split_flag has a first value (e.g., "1"), the current CU may be quadtree-partitioned. In addition, when qt_split_flag has a second value (e.g., "0"), the current CU is not quadtree-partitioned, but becomes the leaf node (QT_leaf_node) of the quadtree. Each quadtree leaf node may then be further partitioned into multitype tree structures. That is, the leaf node of the quadtree may become the node (MTT_node) of the multi-type tree. In the multitype tree structure, a first flag (e.g., Mtt_split_cu_flag) is signaled to indicate whether the current node is additionally partitioned. If the corresponding node is additionally partitioned (e.g., if the first flag is 1), a second flag (e.g., Mtt_split_cu_vertical_flag) may be signaled to indicate the splitting direction. For example, the splitting direction may be a vertical direction if the second flag is 1 and may be a horizontal direction if the second flag is 0. Then, a third flag (e.g., Mtt_split_cu_binary_flag) may be signaled to indicate whether the split type is a binary split type or a ternary split type. For example, the split type may be a binary split type when the third flag is 1 and may be a ternary split type when the third flag is 0. The node of the multi-type tree acquired by binary splitting or ternary splitting may be further partitioned into multi-type tree structures. However, the node of the multi-type tree may not be partitioned into quadtree structures. If the first flag is 0, the corresponding node of the multi-type tree is no longer split but becomes the leaf node (MTT_leaf_node) of the multi-type tree. The CU corresponding to the leaf node of the multi-type tree may be used as the above-described final coding unit.

[120] Based on the mtt_split_cu_vertical_flag and the mtt_split_cu_binary_flag, a multi-type tree splitting mode (MttSplitMode) of a CU may be derived as shown in Table 1 below. In the following description, the multi-tree splitting mode may be referred to as a multi-tree split type or a split type for short.

[121] [Table 1]

MttSplitMode	**mtt_split_cu_vertical_flag**	**mtt_split_cu_binary_flag**
SPLIT_TT_HOR	0	0
SPLIT_BT_HOR	0	1
SPLIT_TT_VER	1	0
SPLIT_BT_VER	1	1

[122] FIG. 7 is a view showing an example in which a CTU is partitioned into multiple CUs by applying a multi-type tree after applying a quadtree. In FIG. 7, bold block edges 710

Date Reçue/Date Received 2023-12-29

represent quadtree partitioning and the remaining edges 720 represent multitype tree partitioning. The CU may correspond to a coding block (CB). In an embodiment, the CU may include a coding block of luma samples and two coding blocks of chroma samples corresponding to the luma samples. A chroma component (sample) CB or TB size may be derived based on a luma component (sample) CB or TB size according to the component ratio according to the color format (chroma format, e.g., 4:4:4, 4:2:2, 4:2:0 or the like) of the picture/image. In case of 4:4:4 color format, the chroma component CB/TB size may be set equal to be luma component CB/TB size. In case of 4:2:2 color format, the width of the chroma component CB/TB may be set to half the width of the luma component CB/TB and the height of the chroma component CB/TB may be set to the height of the luma component CB/TB. In case of 4:2:0 color format, the width of the chroma component CB/TB may be set to half the width of the luma component CB/TB and the height of the chroma component CB/TB may be set to half the height of the luma component CB/TB.

[123] In an embodiment, when the size of the CTU is 128 based on the luma sample unit, the size of the CU may have a size from 128x128 to 4x4 which is the same size as the CTU. In one embodiment, in case of 4:2:0 color format (or chroma format), a chroma CB size may have a size from 64x64 to 2x2.

[124] Meanwhile, in an embodiment, the CU size and the TU size may be the same. Alternatively, there may be a plurality of TUs in a CU region. The TU size generally represents a luma component (sample) transform block (TB) size.

[125] The TU size may be derived based a largest allowable TB size maxTbSize which is a predetermined value. For example, when the CU size is greater than maxTbSize, a plurality of TUs (TBs) having maxTbSize may be derived from the CU and transform/inverse transform may be performed in units of TU (TB). For example, the largest allowable luma TB size may be 64x64 and the largest allowable chroma TB size may be 32x32. If the width or height of the CB partitioned according to the tree structure is larger than the largest transform width or height, the CB may be automatically (or implicitly) partitioned until the TB size limit in the horizontal and vertical directions is satisfied.

[126] In addition, for example, when intra prediction is applied, an intra prediction mode/type may be derived in units of CU (or CB) and a neighboring reference sample derivation and prediction sample generation procedure may be performed in units of TU (or TB). In this case, there may be one or a plurality of TUs (or TBs) in one CU (or CB) region and, in this case, the plurality of TUs or (TBs) may share the same intra prediction mode/type.

[127] Meanwhile, for a quadtree coding tree scheme with a nested multitype tree, the following parameters may be signaled as SPS syntax elements from the image encoding apparatus to the decoding apparatus. For example, at least one of a CTU size which is a parameter representing the root node size of a quadtree, MinQTSize which is a parameter representing the minimum allowed quadtree leaf node size, MaxBtSize which is a parameter

Date Reçue/Date Received 2023-12-29

representing the maximum allowed binary tree root node size, MaxTtSize which is a parameter representing the maximum allowed ternary tree root node size, MaxMttDepth which is a parameter representing the maximum allowed hierarchy depth of multi-type tree splitting from a quadtree leaf node, MinBtSize which is a parameter representing the minimum allowed binary tree leaf node size, or MinTtSize which is a parameter representing the minimum allowed ternary tree leaf node size is signaled.

[128] As an embodiment of using 4:2:0 chroma format, the CTU size may be set to 128x128 luma blocks and two 64x64 chroma blocks corresponding to the luma blocks. In this case, MinOTSize may be set to 16x16, MaxBtSize may be set to 128x128, MaxTtSzie may be set to 64x64, MinBtSize and MinTtSize may be set to 4x4, and MaxMttDepth may be set to 4. Quadtree partitioning may be applied to the CTU to generate quadtree leaf nodes. The quadtree leaf node may be called a leaf QT node. Quadtree leaf nodes may have a size from a 16x16 size (e.g., the MinOTSize) to a 128x128 size (e.g., the CTU size). If the leaf QT node is 128x128, it may not be additionally partitioned into a binary tree/ternary tree. This is because, in this case, even if partitioned, it exceeds MaxBtsize and MaxTtszie (e.g., 64x64). In other cases, leaf QT nodes may be further partitioned into a multitype tree. Therefore, the leaf QT node is the root node for the multitype tree, and the leaf QT node may have a multitype tree depth (mttDepth) 0 value. If the multitype tree depth reaches MaxMttdepth (e.g., 4), further partitioning may not be considered further. If the width of the multitype tree node is equal to MinBtSize and less than or equal to 2xMinTtSize, then no further horizontal partitioning may be considered. If the height of the multitype tree node is equal to MinBtSize and less than or equal to 2xMinTtSize, no further vertical partitioning may be considered. When partitioning is not considered, the image encoding apparatus may skip signaling of partitioning information. In this case, the image decoding apparatus may derive partitioning information with a predetermined value.

[129] Meanwhile, one CTU may include a coding block of luma samples (hereinafter referred to as a "luma block") and two coding blocks of chroma samples corresponding thereto (hereinafter referred to as "chroma blocks"). The above-described coding tree scheme may be equally or separately applied to the luma block and chroma block of the current CU. Specifically, the luma and chroma blocks in one CTU may be partitioned into the same block tree structure and, in this case, the tree structure is represented as SINGLE_TREE. Alternatively, the luma and chroma blocks in one CTU may be partitioned into separate block tree structures, and, in this case, the tree structure may be represented as DUAL_TREE. That is, when the CTU is partitioned into dual trees, the block tree structure for the luma block and the block tree structure for the chroma block may be separately present. In this case, the block tree structure for the luma block may be called DUAL_TREE_LUMA, and the block tree structure for the chroma component may be called DUAL_TREE_CHROMA. For P and B slice/tile groups, luma and chroma blocks in one CTU may be limited to have the same coding

Date Reçue/Date Received 2023-12-29

tree structure. However, for I slice/tile groups, luma and chroma blocks may have a separate block tree structure from each other. If the separate block tree structure is applied, the luma CTB may be partitioned into CUs based on a particular coding tree structure, and the chroma CTB may be partitioned into chroma CUs based on another coding tree structure. That is, this means that a CU in an I slice/tile group, to which the separate block tree structure is applied, may include a coding block of luma components or coding blocks of two chroma components and a CU of a P or B slice/tile group may include blocks of three color components (a luma component and two chroma components).

[130] Although a quadtree coding tree structure with a nested multitype tree has been described, a structure in which a CU is partitioned is not limited thereto. For example, the BT structure and the TT structure may be interpreted as a concept included in a multiple partitioning tree (MPT) structure, and the CU may be interpreted as being partitioned through the QT structure and the MPT structure. In an example where the CU is partitioned through a QT structure and an MPT structure, a syntax element (e.g., MPT_split_type) including information on how many blocks the leaf node of the QT structure is partitioned into and a syntax element (e.g., MPT_split_mode) including information on which of vertical and horizontal directions the leaf node of the QT structure is partitioned into may be signaled to determine a partitioning structure.

[131] In another example, the CU may be partitioned in a different way than the QT structure, BT structure or TT structure. That is, unlike that the CU of the lower depth is partitioned into 1/4 of the CU of the higher depth according to the QT structure, the CU of the lower depth is partitioned into 1/2 of the CU of the higher depth according to the BT structure, or the CU of the lower depth is partitioned into 1/4 or 1/2 of the CU of the higher depth according to the TT structure, the CU of the lower depth may be partitioned into 1/5, 1/3, 3/8, 3/5, 2/3, or 5/8 of the CU of the higher depth in some cases, and the method of partitioning the CU is not limited thereto.

[132] The quadtree coding block structure with the multi-type tree may provide a very flexible block partitioning structure. Because of the partition types supported in a multi-type tree, different partition patterns may potentially result in the same coding block structure in some cases. In the image encoding apparatus and the decoding apparatus, by limiting the occurrence of such redundant partition patterns, a data amount of partitioning information may be reduced.

[133] In addition, in encoding and decoding of a video/image according to the present document, an image processing basis may have a hierarchical structure. One picture may be divided into one or more tiles, bricks, slices, and/or tile groups. One slice may include one or more bricks. One brick may include one or more CTU rows within a tile. A slice may include bricks of a picture, wherein the number of the bricks is an integer. One tile group may include one or more tiles. One tile may include one or more CTUs. The CTU may be partitioned into

Date Reçue/Date Received 2023-12-29

one or more CUs. A tile may be a quadrangular area composed of particular tile rows and particular tile columns composed of a plurality of CTUs within a picture. A tile group may include tiles according to tile raster scanning within a picture, wherein the number of the tiles is an integer. A slice header may carry information/a parameter applicable to the corresponding slice (blocks in the slice). When the encoding apparatus or the decoding apparatus has a multi-core processor, encoding/decoding procedures for the tile, slice, brick, and/or tile group may be performed in parallel.

[134] In the present disclosure, the names or concepts of a slice or a tile group may be used interchangeably. That is, a tile group header may be referred to as a slice header. Herein, a slice may have one of slice types including an intra (I) slice, a predictive (P) slice, and a bi-predictive (B) slice. For blocks within an I slice, inter prediction is not used for prediction, and only intra prediction may be used. Even in this case, an original sample value may be coded and signaled without prediction. For blocks within a P slice, intra prediction or inter prediction may be used. When inter prediction is used, only uni-prediction may be used. In the meantime, for blocks within a B slice, intra prediction or inter prediction may be used. When inter prediction is used, bi-prediction, which is the maximum extent, may be used.

[135] According to a characteristic (for example, resolution) of a video image or considering coding efficiency or parallel processing, the encoding apparatus may determine a tile/tile group, a brick, a slice, and largest and smallest coding unit sizes. In addition, information on this or information for deriving this may be included in a bitstream.

[136] The decoding apparatus may obtain information indicating whether a CTU within a tile/tile group, a brick, a slice, or a tile of a current picture is partitioned into multiple coding units. The encoding apparatus and the decoding apparatus signal such information only under a particular condition, thereby increasing encoding efficiency.

[137] The slice header (slice header syntax) may include information/a parameter applicable to the slice in common. APS (APS syntax) or PPS (PPS syntax) may include information/a parameter applicable to one or more pictures in common. SPS (SPS syntax) may include information/a parameter applicable to one or more sequences in common. VPS (VPS syntax) may include information/a parameter applicable to multiple layers in common. DPS (DPS syntax) may include information/a parameter applicable to the entire video in common. DPS may include information/a parameter related to combination of coded video sequences (CVSs).

[138] In addition, for example, information on the partitioning and configuration of the tile/tile group/brick/slice may be constructed at the encoding stage through the high level syntax, and transmitted in the form of a bitstream to the decoding apparatus.

[139] <u>Overview of intra prediction</u>

[140] Hereinafter, intra prediction performed by the encoding apparatus and the decoding apparatus described above will be described in detail. Intra prediction may refer to prediction

Date Reçue/Date Received 2023-12-29

that generates prediction samples for a current block on the basis of reference samples in a picture (hereinafter, a current picture) to which the current block belongs.

[141] This will be described with reference to FIG. 8. When intra prediction is applied to a current block 801, neighboring reference samples to be used for intra prediction of the current block 801 may be derived. The neighboring reference samples of the current block may include: a total of 2xnH samples including samples 811 adjacent to a left boundary of the current block having a size of nWxnH and samples 812 neighboring the bottom-left; a total of 2xnW samples including samples 821 adjacent to the top boundary of the current block and samples 822 neighboring the top-right; and one sample 831 neighboring the top-left of the current block. Alternatively, the neighboring reference samples of the current block may include a plurality of columns of top neighboring samples and a plurality of rows of left neighboring samples.

[142] In addition, the neighboring reference samples of the current block may include: a total of nH samples 841 adjacent to the right boundary of the current block having a size of nWxnH; a total of nW samples 851 adjacent to the bottom boundary of the current block; and one sample 842 neighboring the bottom-right of the current block.

[143] However, some of the neighboring reference samples of the current block have not yet been decoded or may be unavailable. In this case, the decoding apparatus may construct neighboring reference samples to be used for prediction, by substituting unavailable samples with available samples. Alternatively, neighboring reference samples to be used for prediction may be constructed through interpolation of available samples.

[144] When the neighboring reference samples are derived, (i) a prediction sample may be derived on the basis of an average or interpolation of the neighboring reference samples of the current block, or (ii) the prediction sample may be derived on the basis of the reference sample present in a particular (prediction) direction with respect to the prediction sample, among the neighboring reference samples of the current block. The case of (i) may be referred to as a non-directional mode or a non-angular mode, and the case of (ii) may be referred to as a directional mode or an angular mode. In addition, the prediction sample may be generated through interpolation with the second neighboring sample and the first neighboring sample that are located in the opposite direction of the prediction direction of the intra prediction mode of the current block on the basis of the prediction sample of the current block, among the neighboring reference samples. The above-described case may be referred to as a linear interpolation intra prediction (LIP). In addition, chroma prediction samples may be generated on the basis of luma samples by using a linear model. This case may be referred to as an LM mode. In addition, a temporary prediction sample of the current block may be derived on the basis of filtered neighboring reference samples, and the prediction sample of the current block may be derived by weighted-summing the temporary prediction sample and at least one reference sample derived according to the intra prediction mode among the existing neighboring reference samples, namely, the unfiltered neighboring reference samples. The above-described case may

Date Reçue/Date Received 2023-12-29

be referred to as Position dependent intra prediction (PDPC). In addition, a reference sample line having the highest prediction accuracy may be selected among multiple neighboring reference sample lines of the current block to derive a prediction sample by using a reference sample located in a prediction direction in the corresponding line. At this time, intra prediction encoding may be performed by indicating (signaling) the used reference sample line to the decoding apparatus. The above-described case may be referred to as multi-reference line (MRL) intra prediction or MRL based intra prediction. In addition, the current block may be divided into vertical or horizontal sub-partitions to perform intra prediction based on the same intra prediction mode, and neighboring reference samples may be derived on a per sub-partition basis and used. That is, in this case, the intra prediction mode for the current block is equally applied to the sub-partitions, and a neighboring reference sample is derived on a per sub-partition basis and used, thereby increasing intra prediction performance in some cases. This prediction method may be referred to as intra sub-partitions (ISP) or ISP based intra prediction. These intra prediction methods may be referred to as intra prediction types, being distinguished from intra prediction modes (e.g., a DC mode, a planar mode, and a directional mode). The intra prediction types may be referred to as various terms such as intra prediction schemes or additional intra prediction modes. For example, the intra prediction types (or additional intra prediction modes) may include at least one selected from a group of LIP, PDPC, MRL, and ISP that are described above. A general intra prediction method excluding the particular intra prediction types, such as LIP, PDPC, MRL, and ISP, may be referred to as a normal intra prediction type. The normal intra prediction type may refer to a case in which the particular intra prediction types are not applied, and prediction may be performed on the basis of the above-described intra prediction modes. In the meantime, when necessary, post-filtering may be performed on the derived prediction sample.

[145] Specifically, an intra prediction procedure may include an intra prediction mode/type determination step, a neighboring reference sample derivation step, and an intra prediction mode/type based prediction sample derivation step. In addition, when necessary, a post-filtering step may be performed on the derived prediction sample.

[146] In the meantime, in addition to the above-described intra prediction types, affine linear weighted intra prediction (ALWIP) may be used. The ALWIP may be referred to as linear weighted intra prediction (LWIP), or matrix weighted intra prediction or matrix based intra prediction (MIP). When the MIP is applied to a current block, prediction samples for the current block may be derived by i) using neighboring reference samples subjected to an averaging procedure, ii) performing a matrix-vector-multiplication procedure, and further performing iii) a horizontal/vertical interpolation procedure when necessary. The intra prediction modes used for the MIP may be different from the intra prediction modes used in LIP, PDPC, MRL, ISP intra prediction, or in normal intra prediction. The intra prediction modes for the MIP may be referred to as MIP intra prediction modes, MIP prediction modes, or MIP modes. For example,

Date Reçue/Date Received 2023-12-29

different matrices and offsets used in the matrix-vector-multiplication may be set according to the intra prediction modes for MIP. Herein, a matrix may be referred to as a (MIP) weighted matrix, and an offset may be referred to as an (MIP) offset vector or (MIP) bias vector. A detailed MIP method will be described later.

[147] A block reconstruction procedure based on intra prediction and the intra prediction unit in the encoding apparatus may schematically include, for example, the following described below. Step S910 may be performed by the intra prediction unit 185 of the encoding apparatus. Step S920 may be performed by the residual processor that includes at least one selected from a group of the subtractor 115, the transformer 120, the quantizer 130, the dequantizer 140, and the inverse transformer 150 of the encoding apparatus. Specifically, step S920 may be performed by the subtractor 115 of the encoding apparatus. In step S930, prediction information may be derived by the intra prediction unit 185, and may be encoded by the entropy encoder 190. In step S930, residual information may be derived by the residual processor, and may be encoded by the entropy encoder 190. The residual information is information on the residual samples. The residual information may include information on quantized transformation coefficients for the residual samples. As described above, the residual samples may be derived as transformation coefficients through the transformer 120 of the encoding apparatus, and the transformation coefficients may be derived as quantized transformation coefficients through the quantizer 130. Information on the quantized transformation coefficients may be encoded by the entropy encoder 190 through a residual coding procedure.

[148] The encoding apparatus may perform intra prediction on a current block in step S910. The encoding apparatus derives an intra prediction mode/type for the current block, derives neighboring reference samples of the current block, and generates prediction samples in the current block on the basis of the intra prediction mode/type and the neighboring reference samples. Herein, the procedures of determination of the intra prediction mode/type, derivation of the neighboring reference samples, and generation of the prediction samples may be performed simultaneously, or any one procedure may be performed before the other procedures. For example, although not shown, the intra prediction unit 185 of the encoding apparatus may include an intra prediction mode/type determination unit, a reference sample derivation unit, and a prediction sample derivation unit. The intra prediction mode/type determination unit may determine an intra prediction mode/type for the current block, the reference sample derivation unit may derive neighboring reference samples of the current block, and the prediction sample derivation unit may derive prediction samples of the current block. In the meantime, when a prediction sample filtering procedure, which will be described later, is performed, the intra prediction unit 185 may further include a prediction sample filter. The encoding apparatus may determine a mode/type applied to the current block among a plurality of intra prediction modes/types. The encoding apparatus may compare RD costs for the intra prediction modes/types and determine the optimum intra prediction mode/type for the current block.

Date Reçue/Date Received 2023-12-29

[149] In the meantime, the encoding apparatus may perform a prediction sample filtering procedure. Prediction sample filtering may be referred to as post-filtering. By the prediction sample filtering procedure, some or all of the prediction samples may be filtered. In some cases, the prediction sample filtering procedure may be omitted.

[150] The encoding apparatus may generate residual samples for the current block on the basis of (filtered) prediction samples in step S920. The encoding apparatus may compare the prediction samples in the original samples of the current block on the basis of a phase, and may derive the residual samples.

[151] The encoding apparatus may encode image information including information (prediction information) on the intra prediction and residual information on the residual samples in step S930. The prediction information may include the intra prediction mode information and the intra prediction type information. The encoding apparatus may output encoded image information in the form of a bitstream. The output bitstream may be transmitted to the decoding apparatus through a storage medium or a network.

[152] The residual information may include residual coding syntax, which will be described later. The encoding apparatus may derive quantized transformation coefficients by transforming/quantizing the residual samples. The residual information may include information on the quantized transformation coefficients.

[153] In the meantime, as described above, the encoding apparatus may generate a reconstructed picture (including reconstructed samples and a reconstructed block). To this end, the encoding apparatus may perform dequantization/inverse transformation on the quantized transformation coefficients and derive (modified) residual samples. The reason for performing dequantization/inverse transformation after transformation/quantization of the residual samples is to derive residual samples that are the same as the residual samples derived by the decoding apparatus as described above. The encoding apparatus may generate a reconstructed block including reconstructed samples for the current block, on the basis of the prediction samples and the (modified) residual samples. On the basis of the reconstructed block, a reconstructed picture for the current picture may be generated. As described above, an in-loop filtering procedure may be further applied to the reconstructed picture.

[154] A video/image decoding procedure based on intra prediction and the intra prediction unit in the decoding apparatus may schematically include, for example, the following described below. The decoding apparatus may perform the operation corresponding to the operation performed by the encoding apparatus.

[155] Steps S1010 to S1030 may be performed by the intra prediction unit 265 of the decoding apparatus. Prediction information in step S1010 and residual information in step S1040 may be obtained from a bitstream by the entropy decoder 210 of the decoding apparatus. A residual processor including the dequantizer 220 or the inverse transformer 230 of the decoding apparatus or both may derive residual samples for the current block on the basis of

Date Reçue/Date Received 2023-12-29

the residual information. Specifically, the dequantizer 220 of the residual processor may perform dequantization on the basis of quantized transformation coefficients derived on the basis of the residual information, and may derive transformation coefficients. The inverse transformer 230 of the residual processor may perform inverse transformation on the transformation coefficients and may derive residual samples for the current block. Step S1050 may be performed by the adder 235 or the reconstructor of the decoding apparatus.

[156] Specifically, the decoding apparatus may derive an intra prediction mode/type for a current block on the basis of received prediction information (intra prediction mode/type information) in step S1010. The decoding apparatus may derive neighboring reference samples of the current block in step S1020. The decoding apparatus may generate prediction samples in the current block on the basis of the intra prediction mode/type and the neighboring reference samples in step S1030. In this case, the decoding apparatus may perform a prediction sample filtering procedure. Prediction sample filtering may be referred to as post-filtering. By the prediction sample filtering procedure, some or all of the prediction samples may be filtered. In some cases, the prediction sample filtering procedure may be omitted.

[157] The decoding apparatus may generate residual samples for the current block on the basis of received residual information. The decoding apparatus may generate reconstructed samples for the current block on the basis of the prediction samples and the residual samples, and may derive a reconstructed block including the reconstructed samples in step S1040. On the basis of the reconstructed block, a reconstructed picture for the current picture may be generated. As described above, an in-loop filtering procedure may be further applied to the reconstructed picture.

[158] Herein, although not shown, the intra prediction unit 265 of the decoding apparatus may include an intra prediction mode/type determination unit, a reference sample derivation unit, and a prediction sample derivation unit. The intra prediction mode/type determination unit may determine an intra prediction mode/type for the current block on the basis of the intra prediction mode/type information obtained from the entropy decoder 210. The reference sample derivation unit may derive neighboring reference samples of the current block. The prediction sample derivation unit may derive prediction samples of the current block. In the meantime, when the above-described prediction sample filtering procedure is performed, the intra prediction unit 265 may further include a prediction sample filter.

[159] The intra prediction mode information may include, for example, flag information (e.g., intra_luma_mpm_flag) indicating whether a most probable mode (MPM) is applied to the current block or a remaining mode is applied. When the MPM is applied to the current block, the prediction mode information may further include index information (e.g., intra_luma_mpm_idx) indicating one of the intra prediction mode candidates (MPM candidates). The intra prediction mode candidates (MPM candidates) may be constructed as a MPM candidate list or an MPM list. In addition, when the MPM is not applied to the current

Date Reçue/Date Received 2023-12-29

block, the intra prediction mode information may further include remaining mode information (e.g., intra_luma_mpm_remainder) indicating one of the remaining intra prediction modes except the intra prediction mode candidates (MPM candidates). The decoding apparatus may determine an intra prediction mode of the current block on the basis of the intra prediction mode information. A separate MPM list may be constructed for the above-described MIP.

[160] In addition, the intra prediction type information may be realized in various forms. For example, the intra prediction type information may include intra prediction type index information indicating one of the intra prediction types. As another example, the intra prediction type information may include at least one of reference sample line information (e.g., intra_luma_ref_idx) indicating whether the MRL is applied to the current block and which reference sample line is used when the MRL is applied to the current block, ISP flag information (e.g., intra_subpartitions_mode_flag) indicating whether the ISP is applied to the current block, ISP type information (e.g., intra_subpartitions_split_flag) indicating a partition type of sub-partitions when the ISP is applied, flag information indicating whether PDCP is applied, or flag information indicating whether LIP is applied. In addition, the intra prediction type information may include an MIP flag indicating whether MIP is applied to the current block.

[161] The intra prediction mode information and/or the intra prediction type information may be encoded/decoded through a coding method described in the present document. For example, the intra prediction mode information and/or the intra prediction type information may be encoded/decoded through entropy coding (e.g., CABAC, CAVLC) on the basis of a truncated (rice) binary code.

[162] Overview of inter prediction

[163] Hereinafter, a detailed description of the inter prediction method in the description of encoding and decoding with reference to FIGS. 2 and 3 will be described. In the case of the decoding device, a video/image decoding method based on inter prediction may be performed and the inter prediction unit in the decoding device may operate according to the following description. In the case of the encoding device, a video/image encoding method based on inter prediction may be performed and the inter prediction unit in the encoding device may operate according to the following description. In addition, data encoded according to the following description may be stored in the form of a bitstream.

[164] The prediction unit of the encoding device/decoding device may derive a prediction sample by performing inter prediction on a per-block basis. Inter prediction may refer to prediction that is derived by a method dependent on data elements (e.g., sample values or motion information) of a picture (s) other than a current picture. When inter prediction is applied to a current block, a predicted block (a prediction sample array) for the current block may be derived on the basis of a reference block (a reference sample array) specified by a motion vector on a reference picture indicated by a reference picture index. Herein, in order to

Date Reçue/Date Received 2023-12-29

reduce the amount of motion information transmitted in the inter prediction mode, the motion information of the current block may be predicted on a per block, subblock, or sample basis on the basis of the correlation of motion information between a neighboring block and the current block. The motion information may include a motion vector and a reference picture index. The motion information may further include inter prediction type (L0 prediction, L1 prediction, Bi prediction, etc.) information. When inter prediction is applied, neighboring blocks may include a spatial neighboring block present in the current picture and a temporal neighboring block present in the reference picture. The reference picture including the reference block and the reference picture including the temporal neighboring block may be the same or different. The temporal neighboring block may be called a collocated reference block or a collocated CU (colCU). The reference picture including the temporal neighboring block may be called a collocated picture (colPic). For example, a motion information candidate list may be constructed on the basis of neighboring blocks of the current block. A flag or index information indicating which candidate is selected (used) to derive a motion vector and/or a reference picture index of the current block may be signaled. Inter prediction may be performed on the basis of various prediction modes. For example, in the case of a skip mode and a merge mode, motion information of a current block may be the same as motion information of a selected neighboring block. In the case of the skip mode, unlike the merge mode, the residual signal may not be transmitted. In the case of a motion information prediction (motion vector prediction, MVP) mode, a motion vector of a selected neighboring block may be used as a motion vector predictor, and a motion vector difference may be signaled. In this case, a motion vector of the current block may be derived using the sum of the motion vector predictor and the motion vector difference.

[165] The motion information may include L0 motion information and/or L1 motion information according to an inter prediction type (L0 prediction, L1 prediction, Bi prediction, etc.). A motion vector in the L0 direction may be called an L0 motion vector or MVL0, and a motion vector in the L1 direction may be called an L1 motion vector or MVL1. The prediction based on the L0 motion vector may be called L0 prediction. The prediction based on the L1 motion vector may be called L1 prediction. The prediction based on both the L0 motion vector and the L1 motion vector may be called bi-prediction. Herein, the L0 motion vector may refer to a motion vector associated with a reference picture list L0 (L0), and the L1 motion vector may refer to a motion vector associated with a reference picture list L1 (L1). The reference picture list L0 may include, as reference pictures, pictures preceding the current picture in terms of output order. The reference picture list L1 may include pictures following the current picture in terms of output order. The preceding pictures may be called forward (reference) pictures, and the following pictures may be called backward (reference) pictures. The reference picture list L0 may further include, as reference pictures, pictures following the current picture in terms of output order. In this case, within the reference picture list L0, the preceding pictures may be

Date Reçue/Date Received 2023-12-29

indexed first, and the following pictures may be indexed next. The reference picture list L1 may further include, as reference pictures, pictures preceding the current picture in terms of output order. In this case, within the reference picture list1, the following pictures may be indexed first, and the preceding pictures may be indexed next. Herein, the output order may correspond to the picture order count (POC) order.

[166] A video/image encoding procedure based on inter prediction and the inter prediction unit in the encoding device may schematically include, for example, the following described below. This will be described with reference to FIG. 11. The encoding device may perform inter prediction on a current block in step S1110. The encoding device may derive an inter prediction mode and motion information of the current block, and may generate prediction samples of the current block. Herein, the procedures of determination of the inter prediction mode, derivation of the motion information, and generation of the prediction samples may be performed simultaneously, or any one procedure may be performed before the other procedures. For example, the inter prediction unit of the encoding device may include a prediction mode determination unit, a motion information derivation unit, and a prediction sample derivation unit. The prediction mode determination unit may determine the prediction mode for the current block. The motion information derivation unit may derive the motion information of the current block. The prediction sample derivation unit may derive the prediction samples of the current block. For example, the inter prediction unit of the encoding device may search a predetermined area (search area) of reference pictures for a block similar to the current block through motion estimation, and may derive a reference block of which the difference from the current block is a minimum or less than or equal to a predetermined standard. On the basis of this, a reference picture index indicating the reference picture in which the reference block is located may be derived, and a motion vector may be derived on the basis of the location difference between the reference block and the current block. The encoding device may determine a mode applied to the current block among various prediction modes. The encoding device may compare RD costs for the various prediction modes and may determine the optimum prediction mode for the current block.

[167] For example, when the skip mode or the merge mode is applied to the current block, the encoding device may construct a merge candidate list described later, and may derive a reference block of which the difference from the current block is a minimum or less than or equal to a predetermined standard, among reference blocks indicated by merge candidates included in the merge candidate list. In this case, the merge candidate associated with the derived reference block may be selected, and merge index information indicating the selected merge candidate may be generated and signaled to the decoding device. The motion information of the current block may be derived using the motion information of the selected merge candidate.

Date Reçue/Date Received 2023-12-29

[168] As another example, when the (A)MVP mode is applied to the current block, the encoding device may construct an (A)MVP candidate list described later, and may use, as a motion vector predictor (mvp) of the current block, a motion vector of an mvp candidate selected among mvp candidates included in the (A)MVP candidate list. In this case, for example, a motion vector indicating a reference block derived by the above-described motion estimation may be used as a motion vector of the current block. Among the mvp candidates, an mvp candidate having the motion vector having the minimum difference from the motion vector of the current block may be the selected mvp candidate. A motion vector difference (MVD) that is a difference resulting from subtracting the mvp from the motion vector of the current block may be derived. In this case, information on the MVD may be signaled to the decoding device. In addition, when the (A)MVP mode is applied, a value of the reference picture index may be constructed as reference picture index information and may be separately signaled to the decoding device.

[169] The encoding device may derive residual samples on the basis of the prediction samples in step S1120. The encoding device may compare the prediction samples with the original samples of the current block to derive the residual samples.

[170] The encoding device may encode image information including prediction information and residual information, in step S1130. The encoding device may output the encoded image information in the form of a bitstream. The prediction information is pieces of information related to the prediction procedure, and may include prediction mode information (e.g., a skip flag, a merge flag, or a mode index) and information on motion information. The information on the motion information may include candidate selection information (e.g., a merge index, an mvp flag, or an mvp index) that is information for deriving a motion vector. In addition, the information on the motion information may include the information on the MVD and/or the reference picture index information described above. In addition, the information on the motion information may include information indicating whether L0 prediction, L1 prediction, or bi-prediction is applied. The residual information is information on the residual samples. The residual information may include information on quantized transform coefficients for the residual samples.

[171] The output bitstream may be stored in a (digital) storage medium and transmitted to the decoding device, or may be transmitted to the decoding device over a network.

[172] In the meantime, as described above, the encoding device may generate a reconstructed picture (including reconstructed samples and a reconstructed block) on the basis of the reference samples and the residual samples. This is to derive the same prediction result by the encoding device as that performed in the decoding device, and through this, coding efficiency may be increased. Accordingly, the encoding device may store a reconstructed picture (or reconstructed samples, a reconstructed block) in a memory, and may use the same as a

Date Reçue/Date Received 2023-12-29

reference picture for inter prediction. As described above, an in-loop filtering procedure may be further applied to the reconstructed picture.

[173] A video/image decoding procedure based on inter prediction and the inter prediction unit in the decoding device may schematically include, for example, the following described below.

[174] The decoding device may perform the operation corresponding to the operation performed by the encoding device. The decoding device may perform prediction on the current block on the basis of the received prediction information and may derive prediction samples.

[175] Specifically, the decoding device may determine a prediction mode for the current block on the basis of the received prediction information in step S1210. The decoding device may determine which inter prediction mode is applied to the current block on the basis of the prediction mode information in the prediction information.

[176] For example, on the basis of the merge flag, it may be determined whether the merge mode is applied to the current block or the (A)MVP mode is determined. Alternatively, one of various inter prediction mode candidates may be selected on the basis of the mode index. The inter prediction mode candidates may include a skip mode, a merge mode, and/or an (A)MVP mode, or may include various inter prediction modes described later.

[177] The decoding device may derive motion information of the current block on the basis of the determined inter prediction mode in step S1220. For example, when the skip mode or the merge mode is applied to the current block, the decoding device may construct a merge candidate list described later, and may select one merge candidate among merge candidates included in the merge candidate list. The selection may be performed on the basis of the above-described selection information (a merge index). The motion information of the current block may be derived using the motion information of the selected merge candidate. The motion information of the selected merge candidate may be used as the motion information of the current block.

[178] As another example, when the (A)MVP mode is applied to the current block, the decoding device may construct an (A)MVP candidate list described later, and may use, as a motion vector predictor (mvp) of the current block, a motion vector of an mvp candidate selected among mvp candidates included in the (A)MVP candidate list. The selection may be performed on the basis of the above-described selection information (an mvp flag or an mvp index). In this case, the MVD of the current block may be derived on the basis of the information on the MVD, and the motion vector of the current block may be derived on the basis of the mvp of the current block and the MVD. In addition, a reference picture index of the current block may be derived on the basis of the reference picture index information. A picture indicated by the reference picture index in the reference picture list related to the current block may be derived as a reference picture that is referenced for inter prediction of the current block.

Date Reçue/Date Received 2023-12-29

[179] In the meantime, as will be described later, the motion information of the current block may be derived without constructing a candidate list. In this case, the motion information of the current block may be derived according to a procedure described in a prediction mode described later. In this case, the candidate list configuration as described above may be omitted.

[180] The decoding device may generate prediction samples for the current block on the basis of the motion information of the current block in step S1230. In this case, the reference picture may be derived on the basis of the reference picture index of the current block, and the prediction samples of the current block may be derived using samples of a reference block indicated on the reference picture by the motion vector of the current block. In this case, as described later, in some cases, a prediction sample filtering procedure may be further performed on all or some of the prediction samples of the current block.

[181] For example, the inter prediction unit of the decoding device may include a prediction mode determination unit, a motion information derivation unit, and a prediction sample derivation unit. The prediction mode determination unit may determine the prediction mode for the current block on the basis of received prediction mode information. The motion information derivation unit may derive the motion information (a motion vector and/or a reference picture index) of the current block on the basis of received information on the motion information. The prediction sample derivation unit may derive the prediction samples of the current block.

[182] The decoding device may generate residual samples for the current block on the basis of the received residual information in step S1240. The decoding device may generate reconstructed samples for the current block on the basis of the prediction samples and the residual samples, and may generate, on the basis of this, a reconstructed picture in step S1250. Afterward, as described above, an in-loop filtering procedure may be further applied to the reconstructed picture.

[183] As described above, the inter prediction procedure may include the steps of determining an inter prediction mode, deriving motion information according to the determined prediction mode, and performing prediction based on the derived motion information (generating a prediction sample). The inter prediction procedure may be performed in the encoding device and the decoding device as described above.

[184] <u>Quantization/dequantization</u>

[185] As described above, the quantizer of the encoding apparatus may derive quantized transformation coefficients by applying quantization to transformation coefficients. The dequantizer of the encoding apparatus or the dequantizer of the decoding apparatus may derive transformation coefficients by applying dequantization to quantized transformation coefficients.

[186] In encoding and decoding of a video/still image, a quantization ratio may be changed, and the changed quantization ratio may be used to adjust a compression ratio. From an angle

Date Reçue/Date Received 2023-12-29

on realization, considering complexity, a quantization parameter (QP) may be used instead of using a quantization ratio directly. For example, quantization parameters having integer values of 0 to 63 may be used, and each quantization parameter value may correspond to an actual quantization ratio. In addition, a quantization parameter QP_Y for a luma component (luma sample) and a quantization parameter QP_C for a chroma component (chroma sample) may be set to be different from each other.

[187] In a quantization process, a transformation coefficient C may be input and divided by a quantization ratio Qstep, and on the basis of this, a quantized transformation coefficient C` may be obtained. In this case, considering calculation complexity, a quantization ratio may be multiplied by a scale to be in the form of an integer, and shift operation may be performed by the value corresponding to the scale value. Based on multiplication of a quantization ratio and a scale value, a quantization scale may be derived. That is, the quantization scale may be derived according to a QP. The quantization scale may be applied to the transformation coefficient C, and on the basis of this, the quantized transformation coefficient C` may be derived.

[188] The dequantization process is the reverse process of the quantization process. A quantized transformation coefficient C` may be multiplied by a quantization ratio Qstep, and on the basis of this, a reconstructed transformation coefficient C`` may be obtained. In this case, a level scale may be derived according to the quantization parameter. The level scale may be applied to the quantized transformation coefficient C`, and on the basis of this, a reconstructed transformation coefficient C`` may be derived. The reconstructed transformation coefficient C`` may be slightly different from the original transformation coefficient C because of the loss in a transformation and/or quantization process. Therefore, the encoding apparatus may perform dequantization in the same manner as the decoding apparatus.

[189] In the meantime, an adaptive frequency weighting quantization technology in which a quantization strength is adjusted according to frequency may be applied. The adaptive frequency weighting quantization technology is a method of applying quantization strengths that vary from frequency to frequency. In the adaptive frequency weighting quantization, quantization strengths varying from frequency to frequency may be applied by using a pre-defined quantization scaling matrix. That is, the above-described quantization/dequantization process may be performed further on the basis of the quantization scaling matrix. For example, different quantization scaling matrices may be used according to a size of a current block and/or whether the prediction mode applied to the current block to generate a residual signal of the current block is inter prediction or intra prediction. The quantization scaling matrix may be referred to as a quantization matrix or a scaling matrix. The quantization scaling matrix may be pre-defined. In addition, for frequency adaptive scaling, frequency-specific quantization scale information for the quantization scaling matrix may be constructed/encoded by the encoding apparatus, and may be signaled to the decoding apparatus. The frequency-specific

Date Reçue/Date Received 2023-12-29

quantization scale information may be referred to as quantization scaling information. The frequency-specific quantization scale information may include scaling list data (scaling_list_data). The (modified) quantization scaling matrix may be derived on the basis of the scaling list data. In addition, the frequency-specific quantization scale information may include a present flag information indicating whether the scaling list data is present. Alternatively, when the scaling list data is signaled at a high level (e.g., an SPS), further included is information indicating whether the scaling list data is modified at a lower level (e.g., a PPS or a tile group header, etc.).

[190] Transformation/inverse transformation

[191] As described above, the encoding apparatus may derive a residual block (residual samples) on the basis of a block (prediction samples) predicted through intra/inter/IBC prediction, and may derive quantized transformation coefficients by applying transformation and quantization to the derived residual samples. Being included in residual coding syntax, information (residual information) on quantized transformation coefficients may be encoded and output in the form of a bitstream. The decoding apparatus may obtain information (residual information) on the quantized transformation coefficients from the bitstream, and may derive the quantized transformation coefficients by performing decoding. The decoding apparatus may derive residual samples through dequantization/inverse transformation on the basis of the quantized transformation coefficients. As described above, either the quantization/dequantization or the transformation/inverse transformation or both may be omitted. When the transformation/inverse transformation is omitted, the transformation coefficient may be referred to as a coefficient or a residual coefficient, or may still be referred to as a transformation coefficient for consistency of expression. Whether the transformation/inverse transformation is omitted may be signaled on the basis of a transformation skip flag(e.g., transform_skip_flag).

[192] The transformation/inverse transformation may be performed on the basis of a transformation kernel(s). For example, a multiple transform selection (MTS) scheme for performing transformation/inverse transformation may be applied. In this case, some of multiple transformation kernel sets may be selected and applied to a current block. A transformation kernel may be referred to as various terms, such as a transformation matrix, a transformation type, etc. For example, a transformation kernel set may refer to a combination of a vertical-direction transformation kernel (vertical transformation kernel) and a horizontal-direction transformation kernel (horizontal transformation kernel).

[193] The transformation/inverse transformation may be performed on a per CU or TU basis. That is, the transformation/inverse transformation may be applied to residual samples in a CU or residual samples in a TU. A CU size and a TU size may be the same, or a plurality of TUs may be present in a CU area. In the meantime, a CU size may generally refer to a luma component (sample) CB size. A TU size may generally refer to a luma component (sample)

Date Reçue/Date Received 2023-12-29

TB size. A chroma component (sample) CB or TB size may be derived on the basis of a luma component (sample) CB or TB size according to a component ratio according to a color format (a chroma format, e.g., 4:4:4, 4:2:2, 4:2:0, or the like). The TU size may be derived on the basis of maxTbSize. For example, when the CU size is greater than the maxTbSize, a plurality of TUs (TBs) having the maxTbSize may be derived from the CU and transformation/inverse transformation may be performed on a per TU (TB) basis. The maxTbSize may be considered in determining whether various intra prediction types, such as ISP, are applied. Information on the maxTbSize may be pre-determined. Alternatively, information on the maxTbSize may be generated and encoded by the encoding apparatus and signaled to the encoding apparatus.

[194] Entropy coding

[195] As described above with reference to FIG. 2, some or all of video/image information may be entropy-encoded by the entropy encoder 190. Some or all of video/image information described with reference to FIG. 3 may be entropy-decoded by the entropy decoder 310. In this case, the video/image information may be encoded/decoded on a per syntax element basis. In the present document, encoding/decoding of information may include encoding/decoding by the method described in this paragraph.

[196] FIG. 13 shows a block diagram of CABAC for encoding one syntax element. In an encoding process of CABAC, first, when an input signal is a syntax element not a binary value, the input signal is transformed into a binary value through binarization. When an input signal is already a binary value, the input signal bypasses binarization. Herein, each binary number 0 or 1 constituting a binary value may be referred to as a bin. For example, a binary string (bin string) after binarization is 110, each of 1, 1, and 0 is referred to as one bin. The bin(s) for one syntax element may refer to a value of the syntax element.

[197] Binarized bins may be input to a regular coding engine or a bypass coding engine. The regular coding engine may assign a context model that applies a probability value to a corresponding bin, and may encode the bin on the basis of the assigned context model. After performing coding on each bin, the regular coding engine may update a probability model for the bin. The bins coded in this way may be referred to as context-coded bins. The bypass coding engine may omit a procedure for estimating a probability of an input bin and a procedure for updating the probability model applied to the bin after coding. In the case of the bypass coding engine, an input bin is coded by applying a uniform probability distribution (e.g., 50:50) instead of assigning a context, thereby improving a coding rate. The bins coded in this way may be referred to as bypass bins. The context model may be assigned and updated for each bin to be context-coded (regular-coded), and the context model may be indicated on the basis of ctxidx or ctxInc. ctxidx may be derived on the basis of ctxInc. Specifically, for example, the context index (ctxidx) indicating a context model for each of the regular-coded bins may be derived as the sum of a context index increment (ctxInc) and a context index offset (ctxIdxOffset). Herein, ctxInc varying from bin to bin may be derived. The ctxIdxOffset may be represented by the

Date Reçue/Date Received 2023-12-29

lowest value of the ctxIdx. The lowest value of the ctxIdx may be referred to as an initial value (initValue) of the ctxIdx. The ctxIdxOffset is a value generally used for distinguishment from context models for other syntax elements, and a context model for one syntax element may be distinguished/derived on the basis of ctxinc.

[198] In an entropy encoding procedure, it is determined whether to perform encoding through the regular coding engine or perform encoding through the bypass coding engine, and a coding path may be switched. Entropy decoding may perform the same process as entropy encoding in reverse order.

[199] The above-described entropy coding may be performed as in FIGS. 14 and 15, for example. Referring to FIGS. 14 and 15, the encoding apparatus (entropy encoder) may perform an entropy coding procedure on image/video information. The image/video information may include partitioning-related information, prediction-related information (e.g., inter/intra prediction classification information, intra prediction mode information, and inter prediction mode information), residual information, and in-loop filtering-related information, or may include various syntax elements related thereto. The entropy coding may be performed on a per syntax element basis. Steps S1410 to S1420 of FIG. 14 may be performed by the entropy encoder 190 of the encoding apparatus of FIG. 2 described above.

[200] The encoding apparatus may perform binarization on a target syntax element in step S1410. Herein, the binarization may be based on various binarization methods, such as truncated rice binarization process and fixed-length binarization process, and a binarization method for a target syntax element may be pre-defined. The binarization procedure may be performed by a binarization unit 191 in the entropy encoder 190.

[201] The encoding apparatus may perform entropy encoding on the target syntax element in step S1420. The encoding apparatus may perform regular coding-based (context-based) or bypass coding-based encoding on a bin string of a target syntax element on the basis of an entropy coding scheme such as context-adaptive arithmetic coding (CABAC) or context-adaptive variable length coding (CAVLC). The output may be included in a bitstream. The entropy encoding procedure may be performed by an entropy encoding processor 192 in the entropy encoder 190. As described above, the bitstream may be transmitted to the decoding apparatus through a (digital) storage medium or a network.

[202] Referring to FIGS. 16 and 17, the decoding apparatus (entropy decoder) may decode encoded image/video information. The image/video information may include partitioning-related information, prediction-related information (e.g., inter/intra prediction classification information, intra prediction mode information, and inter prediction mode information), residual information, and in-loop filtering-related information, or may include various syntax elements related thereto. The entropy coding may be performed on a per syntax element basis. Steps S1610 to S1620 may be performed by the entropy decoder 210 of the decoding apparatus of FIG. 3 described above.

Date Reçue/Date Received 2023-12-29

[203] The decoding apparatus may perform binarization on a target syntax element in step S1610. Herein, the binarization may be based on various binarization methods, such as truncated rice binarization process and fixed-length binarization process, and a binarization method for a target syntax element may be pre-defined. The decoding apparatus may derive available bin strings (bin string candidates) for available values of a target syntax element through the binarization procedure. The binarization procedure may be performed by a binarization unit 211 in the entropy decoder 210.

[204] The decoding apparatus may perform entropy decoding on the target syntax element in step S1620. While the decoding apparatus sequentially decodes and parses each bin for the target syntax element from an input bit(s) in a bitstream, and may compare a derived bin string with available bin strings for the syntax element. When the derived bin string is the same as one of the available bin strings, a value corresponding to the bin string may be derived as a value of the syntax element. If not, the next bit in the bitstream is further parsed and the above-described procedure is performed again. Through this process, a start bit or an end bit for particular information (particular syntax element) in a bitstream is not used, but variable-length bits are used to signal the information. Through this, relatively fewer bits are assigned for a low value, thereby increasing overall coding efficiency.

[205] The decoding apparatus may perform context-based or bypass-based decoding on each bin in the bin string from a bitstream on the basis of the entropy coding scheme such as CABAC or CAVLC. The entropy decoding procedure may be performed by an entropy decoding processor 212 in the entropy decoder 210. The bitstream may include various types of information for image/video decoding as described above. As described above, the bitstream may be transmitted to the decoding apparatus through a (digital) storage medium or a network.

[206] In the present document, a table (syntax table) including syntax elements may be used to represent signaling of information from the encoding apparatus to the decoding apparatus. The order of syntax elements in the table including the syntax elements used in the present document may refer to the parsing order of syntax elements from a bitstream. The encoding apparatus may construct and encode a syntax table so that the syntax elements are parsed in the parsing order by the decoding apparatus. The decoding apparatus may parse and decode the syntax elements of the syntax table from a bitstream in the parsing order, and may thus obtain values of the syntax elements.

[207] <u>General image/video coding procedure</u>

[208] In image/video coding, pictures constituting an image/video may be encoded/decoded according to decoding order in a series. The picture order corresponding to the output order of decoded pictures may be set to be different from the decoding order, and on the basis of this, forward prediction as well as backward direction may be performed when inter prediction is performed.

Date Reçue/Date Received 2023-12-29

[209] FIG. 18 shows an example of a schematic picture decoding procedure to which an embodiment(s) of the present document is applicable. In FIG. 18, step S1810 may be performed by the entropy decoder 210 of the decoding apparatus described above with reference to FIG. 3. Step S1820 may be performed by the prediction unit including the intra prediction unit 265 and the inter prediction unit 260. Step S1830 may be performed by the residual processor including the dequantizer 220 and the inverse transformer 230. Step S1840 may be performed by the adder 235. Step S1850 may be performed by the filter 240. Step S1810 may include the information decoding procedure described in the present document. Step S1820 may include the inter/intra prediction procedure described in the present document. Step S1830 may include the residual processing procedure described in the present document. Step S1840 may include the block/picture reconstruction procedure described in the present document. Step S1850 may include the in-loop filtering procedure described in the present document.

[210] Referring to FIG. 18, the picture decoding procedure may schematically include, as described above with reference to FIG. 3, an image/video information acquisition procedure from a bitstream (through decoding) in step S1810, the picture reconstruction procedure in steps S1820 to S1840, and the in-loop filtering procedure for a reconstructed picture in step S1850. The picture reconstruction procedure may be performed on the basis of prediction samples and residual samples that are obtained through the inter/intra prediction in step S1820 and the residual processing in step S1830 (dequantization and inverse transformation of quantized transformation coefficients) process described in the present document. A modified reconstructed picture may be generated through the in-loop filtering procedure for a reconstructed picture generated through the picture reconstruction procedure. The modified reconstructed picture may be output as a decoded picture, and may be stored in the decoded picture buffer or the memory 250 of the decoding apparatus to be used later as a reference picture in the inter prediction procedure when a picture is decoded. In some cases, the in-loop filtering procedure may be omitted. In this case, the reconstructed picture may be output as a decoded picture, and may be stored in the decoded picture buffer or the memory 250 of the decoding apparatus to be used later as a reference picture in the inter prediction procedure when a picture is decoded. As described above, the in-loop filtering procedure in step S1850 may include a deblocking filtering procedure, a sample adaptive offset (SAO) procedure, an adaptive loop filter (ALF) procedure, and/or a bi-lateral filter procedure. Some or all thereof may be omitted. In addition, one or some of the deblocking filtering procedure, the sample adaptive offset (SAO) procedure, the adaptive loop filter (ALF) procedure, and the bi-lateral filter procedure may be sequentially applied, or all of them may be sequentially applied. For example, the deblocking filtering procedure may be applied to a reconstructed picture, and then the SAO procedure may be performed. Alternatively, for example, the deblocking filtering procedure may be applied to a reconstructed picture, and then the ALF procedure may be performed. This may be performed in the same manner as in the encoding apparatus.

Date Reçue/Date Received 2023-12-29

[211] FIG. 19 shows an example of a schematic picture encoding procedure to which an embodiment(s) of the present document is applicable. In FIG. 19, step S1910 may be performed by the prediction unit including the intra prediction unit 185 or the inter prediction unit 180 of the encoding apparatus described above with reference to FIG. 2. Step S1920 may be performed by the residual processor including the transformer 120 and/or the quantizer 130. Step S1930 may be performed by the entropy encoder 190. Step S1910 may include the inter/intra prediction procedure described in the present document. Step S1920 may include the residual processing procedure described in the present document. Step S1930 may include the information encoding procedure described in the present document.

[212] Referring to FIG. 19, the picture encoding procedure may schematically include, as described above with reference to FIG. 2, a procedure for encoding information (e.g., prediction information, residual information, and partitioning information) for picture reconstruction and outputting the information in the form of a bitstream, a procedure for generating a reconstructed picture for a current picture, and a procedure (optional) for applying in-loop filtering to the reconstructed picture. The encoding apparatus may derive (modified) residual samples from quantized transformation coefficients through the dequantizer 140 and the inverse transformer 150, and may generate a reconstructed picture on the basis of the prediction samples that are an output in step S1910, and the (modified) residual samples. The generated reconstructed picture may be the same as the reconstructed picture generated by the decoding apparatus described above. The in-loop filtering procedure may be performed on the reconstructed picture to generate a modified reconstructed picture. The modified reconstructed picture may be stored in the decoded picture buffer or the memory 170. Similarly to the case in the decoding apparatus, the modified reconstructed picture may be used later as a reference picture in the inter prediction procedure when a picture is encoded. As described above, in some cases, some or all of the in-loop filtering procedure may be omitted. When the in-loop filtering procedure is performed, (in-loop) filtering-related information (parameter) may be encoded by the entropy encoder 190 and output in the form of a bitstream. The decoding apparatus may perform the in-loop filtering procedure on the basis of the filtering-related information in the same manner as the encoding apparatus.

[213] Through this in-loop filtering procedure, noises, such as a blocking artifact and a ringing artifact, generating during image/video coding may be reduced, and subjective/objective visual quality may be improved. In addition, both the encoding apparatus and the decoding apparatus perform the in-loop filtering procedure, so that the encoding apparatus and the decoding apparatus may derive the same prediction result, the reliability of picture coding may be increased, and the amount of data to be transmitted for picture coding may be reduced.

[214] As described above, the picture reconstruction procedure may be performed in the decoding apparatus as well as the encoding apparatus. A reconstructed block may be generated

Date Reçue/Date Received 2023-12-29

on the basis of intra prediction/inter prediction based on each block basis, and a reconstructed picture including reconstructed blocks may be generated. When a current picture/slice/tile group is an I picture/slice/tile group, the blocks included in the current picture/slice/tile group may be reconstructed on the basis of only intra prediction. In the meantime, when a current picture/slice/tile group is a P or B picture/slice/tile group, the blocks included in the current picture/slice/tile group may be reconstructed on the basis of intra prediction or inter prediction. In this case, inter prediction may be applied to some blocks in the current picture/slice/tile group, and intra prediction may be applied to some remaining blocks. A color component of a picture may include a luma component and a chroma component. Unless explicitly limited in the present document, the method and the embodiments proposed in the present document may be applied to the luma component and the chroma component.

[215] Example of coding hierarchy and structure

[216] A coded video/image according to the present document may be processed according to, for example, a coding hierarchy and structure to be described later.

[217] FIG. 20 is a view showing a hierarchical structure for a coded image. A coded image may be divided into a video coding layer (VCL, video coding hierarchy) that handles decoding processing of an image and itself, a subsystem for transmitting and storing encoded information, and a network abstraction layer (NAL, network abstraction hierarchy) that is present between the VCL and the subsystem, and is in charge of a network adaptation function.

[218] In the VCL, VCL data including compressed image data (slice data) may be generated. Alternatively, a parameter set including information, such as a picture parameter set (PPS), a sequence parameter set (SPS), and a video parameter set (VPS), or a supplemental enhancement information (SEI) message additionally required for an image decoding process may be generated.

[219] In the NAL, header information (NAL unit header) is added to a raw byte sequence payload (RBSP) generated in the VCL so that an NAL unit may be generated. Herein, the RBSP refers to slice data, a parameter set, and an SEI message generated in the VCL. The NAL unit header may include NAL unit type information that is specified according to RBSP data included in the NAL unit.

[220] As shown in the figure, the NAL unit may be divided into a VCL NAL unit and a non-VCL NAL unit depending on the RBSP generated in the VCL. The VCL NAL unit may refer to an NAL unit that includes information (slice data) on an image. The non-VCL NAL unit may refer to an NAL unit that includes information (a parameter set or an SEI message) required for decoding an image.

[221] With header information attached according to a data standard of the subsystem, the VCL NAL unit and the non-VCL NAL unit may be transmitted over a network. For example, an NAL unit may be transformed in the form of data of a predetermined standard, such as

Date Reçue/Date Received 2023-12-29

H.266/VVC file format, a real-time transport protocol (RTP), a transport stream (TS), and the resulting data may be transmitted over various networks.

[222] As described above, regarding an NAL unit, an NAL unit type may be specified according to an RBSP data structure included in the NAL unit, and information on the NAL unit type may be stored in an NAL unit header and signaled.

[223] For example, depending on whether an NAL unit includes information (slice data) on an image, rough classification into a VCL NAL unit type and a non-VCL NAL unit type is made. The VCL NAL unit type may be classified according to a characteristic and a type of a picture included in the VCL NAL unit, and the non-VCL NAL unit type may be classified according to the type of a parameter set.

[224] NAL unit types specified according to types of parameter sets which non-VCL NAL unit types include are listed below as an example.

[225] - APS (Adaptation Parameter Set) NAL unit : a type of an NAL unit including an APS

[226] - DPS (Decoding Parameter Set) NAL unit : a type of an NAL unit including a DPS

[227] - VPS (Video Parameter Set) NAL unit : a type of an NAL unit including a VPS

[228] - SPS (Sequence Parameter Set) NAL unit: a type of an NAL unit including an SPS

[229] - PPS (Picture Parameter Set) NAL unit : a type of an NAL unit including a PPS

[230] The above-described NAL unit types have syntax information for an NAL unit type, and the syntax information may be stored in the NAL unit header and signaled. For example, the syntax information may be nal_unit_type, and NAL unit types may be specified by nal_unit_type values.

[231] The slice header (slice header syntax) may include information/a parameter applicable to the slice in common. The APS (APS syntax) or the PPS (PPS syntax) may include information/a parameter applicable to one or more slices or pictures in common. The SPS (SPS syntax) may include information/a parameter applicable to one or more sequences in common. The VPS (VPS syntax) may include information/a parameter applicable to multiple layers in common. The DPS (DPS syntax) may include information/a parameter applicable to the entire video in common. The DPS may include information/a parameter related to concatenation of coded video sequences (CVSs). In the present document, a high level syntax (HLS) may include at least one selected from the group of the APS syntax, the PPS syntax, the SPS syntax, the VPS syntax, the DPS syntax, and the slice header syntax.

[232] In the present document, image/video information encoded by the encoding apparatus and signaled in the form of a bitstream to the decoding apparatus may include partitioning-related information, intra/inter prediction information, residual information, and in-loop filtering information within a picture as well as information included in the slice header, information included in the APS, information included in the PPS, information included in the SPS, and/or information included in the VPS.

[233] **Overview of adaptive color transform (ACT)**

Date Reçue/Date Received 2023-12-29

[234] Adaptive color transform (ACT) is a color space transformation (conversion) technology for removing unnecessary overlap between color components, and has been used in an HEVC screen content extension version. This may also be applied to VVC.

[235] In the HEVC screen content extension (HEVC SCC extension), ACT has been used to adaptively transform a prediction residual from an existing color space to a YCgCo color space. One of the two color spaces may be optionally selected by signaling one ACT flag for each transformation basis.

[236] For example, a first value (e.g., 1) of the flag may indicate that a residual of the transformation basis is encoded in the original color space. A second value (e.g., 1) of the flag may indicate that a residual of the transformation basis is encoded in the YCgCo color space.

[237] FIG. 21 is a view showing an embodiment of a decoding process to which ACT is applied. In an embodiment of FIG. 21, motion compensated prediction may correspond to inter prediction in the present disclosure.

[238] As shown in FIG. 21, a reconstructed picture (or a reconstructed block, a reconstructed sample array, a reconstructed sample(s), a reconstructed signal) may be generated on the basis of a prediction output value and a residual output value. Herein, the residual output value may be an inverse transformation output value. Herein, inverse transformation may be normal inverse transformation. Herein, the normal inverse transformation may be MTS-based inverse transformation or inverse low frequency non-separable transform (LFNST).

[239] Herein, the prediction output value may be a prediction block, a prediction sample array, a prediction sample(s) or a prediction signal. The residual output value may be a residual block, a residual sample array, a residual sample(s), or a residual signal.

[240] For example, in terms of the encoding apparatus, an ACT process may be performed on residual samples derived on the basis of prediction samples. In addition, an output value of the ACT process may be provided as an input of a normal transformation process. Herein, the normal transformation process may be MTS-based transformation or LFNST.

[241] Information (parameter) on (inverse) ACT may be generated and encoded by the encoding apparatus, and may be transmitted to the decoding apparatus in the form of a bitstream.

[242] The decoding apparatus may obtain, parse, and decode (inverse) ACT-related information (parameter), and may perform inverse ACT on the basis of the (inverse) ACT-related information (parameter).

[243] On the basis of the inverse ACT, (modified) residual samples (or residual block) may be derived. For example, (transformation) coefficients may be derived by applying dequantization to quantized (transformation) coefficients. In addition, residual samples may be derived by performing inverse transformation on (transformation) coefficients. In addition, (modified) residual samples may be obtained by applying inverse ACT to residual samples. The information (parameter) on (inverse) ACT will be described in detail later.

Date Reçue/Date Received 2023-12-29

[244] In an embodiment, a core transformation function used in HEVC may be used as a core transformation function (transformation kernel) for color space transformation. For example, matrices for forward transformation and backward transformation as shown in the equations below may be used.

[245] [Equation 1]

[246] $$\begin{bmatrix} C_0' \\ C_1' \\ C_2' \end{bmatrix} = \begin{bmatrix} 2 & 1 & 1 \\ 2 & -1 & -1 \\ 0 & -2 & 2 \end{bmatrix} \begin{bmatrix} C_0 \\ C_1 \\ C_2 \end{bmatrix} /4$$

[247] [Equation 2]

[248] $$\begin{bmatrix} C_0 \\ C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & -1 & -1 \\ 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} C_0' \\ C_1' \\ C_2' \end{bmatrix}$$

[249] Herein, C0, C1, and C2 may correspond to G, B, and R. Herein, G denotes a green color component, B denotes a blue color component, and R denotes a red color component. In addition, C0', C1', and C2' may correspond to Y, Cg, and Co. Herein, Y denotes a luma, Cg denotes a green chroma, and Co denotes an orange chroma component.

[250] In addition, in order to compensate for a dynamic range change of a residual before and after color transformation, QP adjustment may be applied to a transformation residual by (-5, -5, -3). Details of QP adjustment will be described later.

[251] In the meantime, in the encoding and decoding processes according to an embodiment, when ACT is applicable, the following limitations may be applied.

[252] - In the case of dual tree encoding/decoding, ACT is deactivated. For example, ACT may be applied only to single-tree encoding/decoding.

[253] - When ISP encoding and decoding are applied, ACT may be deactivated.

[254] - For the chroma block to which BDPCM is applied, ACT may be deactivated. Only for the luma block to which BDPCM is applied, ACT may be activated.

[255] - When application of ACT is possible, the CCLM may be deactivated.

[256] FIG. 22 is a view showing an embodiment of a sequence parameter set syntax table in which a syntax element related to ACT is signaled.

[257] FIGS. 23 to 29 are views successively showing an embodiment of a syntax table of an encoding basis in which a syntax element related to ACT is signaled.

[258] As shown in FIG. 22, as an ACT activation flag indicating whether ACT is activated in the decoding process, sps_act_enabled_flag 2210 may be used.

[259] A first value (e.g., 0) of the sps_act_enabled_flag may indicate that ACT is not used and a flag cu_act_enabled_flag 2310, 2910 indicating whether ACT is applied in the encoding basis is not provided in the syntax for the encoding basis.

[260] A second value (e.g., 1) of the sps_act_enabled_flag may indicate that ACT may be used and the cu_act_enabled_flag may be provided in the syntax for the encoding basis.

Date Reçue/Date Received 2023-12-29

[261] When the sps_act_enabled_flag is not obtained in a bitstream, the value of the sps_act_enabled_flag may be derived as the first value (e.g., 0).

[262] In addition, as shown in FIG. 23, as ACT flags indicating whether a residual of the current encoding basis is encoded in the YCgCo color space, the cu_act_enabled_flag 2310, 2910 may be used.

[263] A first value (e.g., 0) of the cu_act_enabled_flag may indicate that a residual of the current encoding basis is encoded in the original color space. A second value (e.g., 1) of the cu_act_enabled_flag may indicate that a residual of the current encoding basis is encoded in the YCgCo color space.

[264] When the cu_act_enabled_flag is not provided in a bitstream, the flag may be derived as the first value (e.g., 0). Herein, the original color space may be an RGB color space.

[265] QP derivation method of transformation basis by using ACT QP offset

[266] In an embodiment, in a scaling process for a transformation coefficient, a derivation process of a quantization parameter and a Qp update process may be performed as follows. For example, the quantization parameter derivation process may be performed using the following parameters.

[267] - Luma coordinates (xCb, yCb) indicating relative coordinates of the top left luma sample of a current encoding block with respect to the top left luma sample of a current picture,

[268] - A variable cbWidth indicating the width of the current encoding block on a per luma sample basis,

[269] - A variable cbHeight indicating the height of the current encoding block on a per luma sample basis

[270] - A variable treeType indicating whether a single tree (SINGLE_TREE) or a dual tree is used to partition a current coding tree node, and indicating, when the dual tree is used, whether the dual tree is a luma component dual tree (DAUL_TREE_LUMA) or a chroma component dual tree (DAUL_TREE_CHROMA)

[271] In the present process, a luma quantization parameter Qp'Y, chroma quantization parameters Qp'Cb, Qp'Cr, and Qp'CbCr may be derived.

[272] A variable luma location (xQg, yQg) may indicate the location of the top left luma sample of a current quantization group corresponding to the top left sample of a current picture. Herein, the horizontal location xQg and the vertical location yQg may be set to be equal to the values of a variable CuQgTopLeftX and a variable CuQgTopLeftY, respectively. The variables CuQgTopLeftX and CuQgTopLeftY may be defined as predetermined values in the coding tree syntax as shown in FIG. 30.

[273] Herein, the current quantization group may be a quadrangular area within a coding tree block, and may share the same qP_{Y_PRED} value. The width and the height thereof may be equal to the width and the height of a coding tree node in which a top left luma sample location is assigned to each of the CuQgTopLeftX and the CuQgTopLeftY.

Date Reçue/Date Received 2023-12-29

[274] When the treeType is the SINGLE_TREE or the DUAL_TREE_LUMA, a luma quantization parameter prediction value qP_{Y_PRED} may be derived as in following steps.

[275] 1. The variable qP_{Y_PRED} may be derived as follows.

[276] (Condition 1) When any one of the following conditions is true, the value of the qP_{Y_PRED} may be set to the same value as $SliceQp_Y$(herein, $SliceQp_Y$ indicates an initial value of a quantization parameter Qp_Y for all slices in a picture, and this may be obtained from a bitstream). Alternatively, the value of the qP_{Y_PRED} may be set to the value of the luma quantization parameter Qp_Y of the last luma encoding basis of the immediately preceding quantization group according to the decoding order.

[277] - (Condition 1-1) When the current quantization group is the first quantization group in a slice

[278] - (Condition 1-2) When the current quantization group is the first quantization group in a tile

[279] - (Condition 1-3) When the current quantization group is the first quantization group in a CTB row in a tile and predetermined synchronization occurs (e.g., when entropy_coding_sync_enabled_flag has a value of 1)

[280] 2. A value of a variable qP_{Y_A} may be derived as follows.

[281] (Condition 2) When at least one of the following conditions is true, the value of the qP_{Y_A} may be set to the value of the qP_{Y_PRED}. Alternatively, the value of the qP_{Y_A} may be set to the value of the luma quantization parameter Qp_Y of the encoding basis that includes a luma encoding block covering a luma sample location (xQg-1, yQg).

[282] - (Condition 2-1) With respect to a block identified by a sample location (xCb, yCb), a block identified by a sample location (xQg - 1, yQg) is not an available neighboring block,

[283] - (Condition 2-2) When a CTB including a luma encoding block covering a luma sample location (xQg - 1, yQg) is different from a CTB including a current luma encoding block at a luma sample location (xCb, yCb), for example, when all the following condition are true

[284] - (Condition 2-2-1) The value of (xQg - 1) >> CtbLog2SizeY is different from that of (xCb) >> CtbLog2SizeY

[285] - (Condition 2-2-2) The value of (yQg) >> CtbLog2SizeY is different from that of (yCb) >> CtbLog2SizeY

[286] 3. A value of a variable qP_{Y_B} may be derived as follows.

[287] (Condition 3) When at least one of the following conditions is true, the value of the qP_{Y_B} may be set to the value of the qP_{Y_PRED}. Alternatively, the value of the qP_{Y_B} may be set to the value of the luma quantization parameter Qp_Y of the encoding basis that includes a luma encoding block covering a luma sample location (xQg, yQg-1).

Date Reçue/Date Received 2023-12-29

[288] - (Condition 3-1) With respect to a block identified by a sample location (xCb, yCb), when a block identified by a sample location (xQg, yQg-1) is not an available neighboring block,

[289] - (Condition 3-2) When a CTB including a luma encoding block covering a luma sample location (xQg, yQg- 1) is different from a CTB including a current luma encoding block at a luma sample location (xCb, yCb), for example, when all the following conditions are true

[290] - (Condition 3-2-1) The value of (xQg) >> CtbLog2SizeY is different from that of (xCb) >> CtbLog2SizeY

[291] - (Condition 3-2-2) The value of (yQg – 1) >> CtbLog2SizeY is different from that of (yCb) >> CtbLog2SizeY

[292] 4. A luma quantization parameter prediction value qP_{Y_PRED} may be derived as follows.

[293] When all the following conditions are true, the qP_{Y_PRED} may be set to the luma quantization parameter Qp_Y of the encoding basis that includes a luma encoding block covering a luma sample location (xQg, yQg - 1).

[294] - (Condition 3-1) With respect to a block identified by a sample location (xCb, yCb), a block identified by a sample location (xQg, yQg-1) is an available neighboring block

[295] - When the current quantization group is the first quantization group in a CTB row in a tile

[296] In the meantime, when all the conditions are not true, the qP_{Y_PRED} may be derived as shown in the equation below.

[297] [Equation 3]

[298] $qP_{Y_PRED} = (qP_{Y_A} + qP_{Y_B} + 1) >> 1$

[299] The variable Qp_Y may be derived as shown in the equation below.

[300] [Equation 4]

[301] $Qp_Y = ((qP_{Y_PRED} + CuQpDeltaVal + 64 + 2 * QpBdOffset) \% (64 + QpBdOffset)) - QpBdOffset$

[302] Herein, the CuQpDeltaVal indicates the difference between a luma quantization parameter for the encoding basis and a prediction value thereof. The value thereof may be obtained from a bitstream. The QpBdOffset indicates a luma and chroma quantization parameter range offset. The QpBdOffset may be preset to a predetermined constant or obtained from a bitstream. For example, the QpBdOffset may be calculated by multiplying a predetermined constant (e.g., 6) by a value of a syntax element (e.g., sps_bitdepth) indicating a bit depth of a luma or chroma sample. The luma quantization parameter Qp'_Y may be derived as shown in the equation below.

[303] [Equation 5]

[304] $Qp'_Y = Qp_Y + QpBdOffset$

Date Reçue/Date Received 2023-12-29

[305] When the value of a variable ChromaArrayType indicating a type of a chroma array is not a first value (e.g., 0) and the treeType is the SINGLE_TREE or the DUAL_TREE_CHROMA, the following processing may be performed.

[306] - When the value of the treeType is the DUAL_TREE_CHROMA, the value of the variable Qp_Y may be set to the same value as that of the luma quantization parameter Qp_Y of the luma encoding basis covering the luma sample location (xCb + cbWidth / 2, yCb + cbHeight / 2).

[307] - Variables qP_{Cb}, qP_{Cr}, and qP_{CbCr} may be derived as shown in the equation below.

[308] [Equation 6]

[309] $qP_{Chroma} = Clip3(-QpBdOffset, 63, Qp_Y)$

[310] $qP_{Cb} = ChromaQpTable[0][qP_{Chroma}]$

[311] $qP_{Cr} = ChromaQpTable[1][qP_{Chroma}]$

[312] $qP_{CbCr} = ChromaQpTable[2][qP_{Chroma}]$

[313] The chroma quantization parameters Qp'_{Cb} and Qp'_{Cr} for Cb and Cr components and the chroma quantization parameter Qp'_{CbCr} for joint Cb-Cr coding may be derived as shown in the equation below.

[314] [Equation 7]

[315] $Qp'_{Cb} = Clip3(-QpBdOffset, 63, qP_{Cb} + pps_cb_qp_offset + slice_cb_qp_offset + CuQpOffset_{Cb}) + QpBdOffset$

[316] $Qp'_{Cr} = Clip3(-QpBdOffset, 63, qP_{Cr} + pps_cr_qp_offset + slice_cr_qp_offset + CuQpOffset_{Cr}) + QpBdOffset$

[317] $Qp'_{CbCr} = Clip3(-QpBdOffset, 63, qP_{CbCr} + pps_joint_cbcr_qp_offset + slice_joint_cbcr_qp_offset + CuQpOffset_{CbCr}) + QpBdOffset$

[318] In the equation above, the pps_cb_qp_offset and the pps_cr_qp_offset are offsets used to derive the Qp'_{Cb} and the Qp'_{Cr}, and may be obtained from a bitstream for a picture parameter set. The slice_cb_qp_offset and the slice_cr_qp_offset are offsets used to derive the Qp'_{Cb} and the Qp'_{Cr}, and may be obtained from a bitstream for a slice header. The $CuQpOffset_{Cb}$ and the $CuQpOffset_{Cr}$ are offsets used to derive the Qp'_{Cb} and the Qp'_{Cr}, and may be obtained from a bitstream for the transformation basis.

[319] In addition, for example, a dequantization process for a transformation coefficient may be performed using the following parameters.

[320] - Luma coordinates (xTbY, yTbY) referring to relative coordinates of the top left sample of the current luma transformation block with respect to the top left luma sample of the current picture

[321] - A variable nTbW indicating the width of a transformation block

Date Reçue/Date Received 2023-12-29

[322] - A variable nTbH indicating the height of a transformation block

[323] - A variable cIdx indicating a color component of a current block

[324] The output of the present process may be an array d of scaled transformation coefficients. Herein, the size of the array d may be (nTbW)x(nTbH). The individual elements constituting this may be identified as d[x][y].

[325] To this end, a quantization parameter qP may be derived as follows. When the cIdx has a value of 0, the qP may be derived as shown in the equation below.

[326] [Equation 8]

[327] $qP = Qp'_{Y}$

[328] Alternatively, when the TuCResMode[xTbY][yTbY] has a value of 2, derivation may be made as shown in the equation below.

[329] [Equation 9]

[330] $qP = Qp'_{CbCr}$

[331] Alternatively, when the cIdx has a value of 1, the qP may be derived as shown in the equation below.

[332] [Equation 10]

[333] $qP = Qp'_{Cb}$

[334] Alternatively, when the cIdx has a value of 2, the qP may be derived as shown in the equation below.

[335] [Equation 11]

[336] $qP = Qp'_{Cr}$

[337] Afterward, the quantization parameter qP may be updated as follows. In addition, variables rectNonTsFlag and bdShift may be derived as follows. For example, when transform_skip_flag[xTbY][yTbY][cIdx] has a value of 0 (e.g., when transformation of a current transformation block is not skipped), derivation may be made as shown in the equation below. Herein, a first value (e.g., 0) of the transform_skip_flag may indicate that whether transformation is omitted is determined by another syntax element. A second value (e.g., 1) of the transform_skip_flag may indicate transformation omission (e.g., skip).

[338] [Equation 12]

[339] qP = qP - (cu_act_enabled_flag[xTbY][yTbY] ? 5 : 0)

[340] rectNonTsFlag = 0

[341] bdShift = 10

[342] Alternatively, when the transform_skip_flag[xTbY][yTbY][cIdx] has a value of 1 (e.g., transformation of the current transformation block is skipped), derivation may be made as shown in the equation below.

[343] [Equation 13]

[344] qP = Max(QpPrimeTsMin, qP) - (cu_act_enabled_flag[xTbY][yTbY] ? 5 : 0)

[345] rectNonTsFlag = (((Log2(nTbW) + Log2(nTbH)) & 1) == 1

Date Reçue/Date Received 2023-12-29

[346] bdShift = BitDepth + (rectNonTsFlag ? 1 : 0) + ((Log2(nTbW) + Log2(nTbH)) / 2) - + pic_dep_quant_enabled_flag

[347] Herein, the QpPrimeTsMin may indicate a minimum quantization parameter value allowed when a transformation skip mode is applied. This may be determined to be a predetermined constant or may be derived from a syntax element of the bitstream related thereto.

[348] Herein, the suffixes Y, Cb, and Cr may denote G, B, and R color components in an RGB color model, or Y, Cg, and Co color components in a YCgCo color model.

[349] <u>Overview of block difference pulse code modulation (BDPCM)</u>

[350] The image encoding apparatus and the image decoding apparatus according to an embodiment may perform differential encoding of a residual signal. For example, the image encoding apparatus may encode the residual signal by subtracting a prediction signal from the residual signal of a current block, and the image decoding apparatus may decode the residual signal by adding the prediction signal to the residual signal of the current block. The image encoding apparatus and the image decoding apparatus according to an embodiment may perform differential encoding of the residual signal by applying BDPCM described below.

[351] BDPCM according to the present disclosure may be performed in a quantized residual domain. The quantized residual domain may include a quantized residual signal (or quantized residual coefficient), and, when applying BDPCM, transform of the quantized residual signal may be skipped. For example, when applying BDPCM, transform of the residual signal may be skipped and quantization may be performed. Alternatively, the quantized residual domain may include quantized transform coefficients.

[352] In an embodiment to which BDPCM applies, the image encoding apparatus may derive a residual block of a current block predicted in an intra prediction mode and quantize the residual block, thereby deriving a residual block. When a differential encoding mode of the residual signal is performed with respect to the current block, the image encoding apparatus may perform differential encoding with respect to the residual block to derive a modified residual block. In addition, the image encoding apparatus may encode differential encoding mode information specifying the differential encoding mode of the residual signal and the modified residual block, thereby generating a bitstream.

[353] More specifically, when BDPCM applies to the current block, a predicted block (prediction block) including predicted samples of the current block may be generated by intra prediction. In this case, an intra prediction mode for performing intra prediction may be signaled through a bitstream and may be derived based on a prediction direction of BDPCM described below. In addition, in this case, the intra prediction mode may be determined to be one of a vertical prediction direction mode or a horizontal prediction direction mode. For example, when the prediction direction of BDPCM is a horizontal direction, the intra prediction mode may be determined to be a horizontal prediction direction mode, and the prediction block

Date Reçue/Date Received 2023-12-29

of the current block may be generated by intra prediction of the horizontal direction. Alternatively, when the prediction direction of BDPCM is a vertical direction, the intra prediction mode may be determined to be a vertical prediction direction mode, and the prediction block of the current block may be generated by intra prediction of the vertical direction. When applying intra prediction of the horizontal direction, a value of a pixel adjacent to the left of the current block may be determined to be a prediction sample value of samples included in a corresponding row of the current block. When applying intra prediction of the vertical direction, a value of a pixel adjacent to the top of the current block may be determined to be a prediction sample value of samples included in a corresponding column of the current block. When applying BDPCM to the current block, a method of generating the prediction block of the current block may be equally performed in an image encoding apparatus and an image decoding apparatus.

[354] When applying BDPCM to the current block, the image encoding apparatus may generate a residual block including residual samples of the current block, by subtracting the prediction sample from the current block. The image encoding apparatus may quantize the residual block and then encode a difference (or delta) between a quantized residual sample and a predictor of the quantized residual sample. The image decoding apparatus may generate the quantized residual block of the current block, by obtaining the quantized residual sample of the current block based on the predictor and the difference reconstructed from a bitstream. Thereafter, the image decoding apparatus may dequantize the quantized residual block and then add it to the prediction block, thereby reconstructing the current block.

[355] FIG. 31 is a view illustrating a method of encoding a residual sample of BDPCM according to the present disclosure. The residual block of FIG. 31 may be generated by subtracting a prediction block from a current block in an image encoding apparatus. The quantized residual block of FIG. 31 may be generated by quantizing the residual block by the image encoding apparatus. In FIG. 31, $r_{i,j}$ specifies a value of a residual sample of the (i, j) coordinates in a current block. When the size of the current block is MxN, a value i may be from 0 to M-1, inclusive. In addition, a value j may be from 0 to N-1, inclusive. For example, a residual may refer to a difference between an original block and a prediction block. For example, $r_{i,j}$ may be derived by subtracting the value of the prediction sample from the value of an original sample of the (i, j) coordinates in the current block. For example, $r_{i,j}$ may be a prediction residual after horizontal intra prediction or vertical intra prediction is performed using a sample that is not filtered from a top or left boundary sample. In the horizontal intra prediction, a value of a left neighboring pixel is copied along a line crossing a prediction block. In the vertical intra prediction, a top neighboring line is copied to an individual line of a prediction block.

[356] In FIG. 31, $Q(r_{i,j})$ refer to a value of a quantized residual sample of coordinates (i, j) in the current block. For example, $Q(r_{i,j})$ may refer to a quantized value of $r_{i,j}$.

Date Reçue/Date Received 2023-12-29

[357] Prediction of BDPCM is performed on the quantized residual samples of FIG. 31, and a modified quantized residual block R' having a MxN size including modified quantized residual samples r' may be generated.

[358] When a prediction direction of BDPCM is a horizontal direction, a value $r'_{i,j}$ of a modified quantized residual sample of coordinates (i, j) in the current block may be calculated as shown in the equation below.

[359] [Equation 14]

[360] $$r'_{i,j} = \begin{cases} Q(r_{i,j}), & i = 0,\ \ 0 \le j \le (N-1) \\ Q(r_{i,j}) - Q(r_{(i-1),j}), & 1 \le i \le (M-1),\ \ 0 \le j \le (N-1) \end{cases}$$

[361] As shown in Equation 14, when the prediction direction of BDPCM is a horizontal direction, a value $Q(r_{0,j})$ of a quantized residual sample is assigned as it is to a value $r'_{0,j}$ of coordinates (0, j). A value $r'_{i,j}$ of other coordinates (i, j) may be derived as a difference value between a value $Q(r_{i,j})$ of a quantized residual sample of coordinates (i, j) and a value $Q(r_{i-1,j})$ of a quantized residual sample of coordinates (i-1, j). That is, instead of encoding a value $Q(r_{i,j})$ of a quantized residual sample of coordinates (i, j), a difference value calculated by using a value $Q(r_{i-1,j})$ of a quantized residual sample of coordinates (i-1, j) as a prediction value is derived as a modified quantized residual sample value $r'_{i,j}$, and then the value $r'_{i,j}$ is encoded.

[362] When a prediction direction of BDPCM is a vertical direction, a value ($r'_{i,j}$) of a modified quantized residual sample of coordinates (i, j) in the current block may be calculated as shown in the equation below.

[363] [Equation 15]

[364] $$r'_{i,j} = \begin{cases} Q(r_{i,j}), & 0 \le i \le (M-1),\ j = 0 \\ Q(r_{i,j}) - Q(r_{i,(j-1)}), & 0 \le i \le (M-1),\ \ 1 \le j \le (N-1) \end{cases}$$

[365] As shown in Equation 15, when the prediction direction of BDPCM is a vertical direction, a value $Q(r_{i,0})$ of a quantized residual sample is assigned as it is to a value $r'_{i,0}$ of coordinates (i, 0). A value $r'_{i,j}$ of other coordinates (i, j) may be derived as a difference value between a value $Q(r_{i,j})$ of a quantized residual sample of coordinates (i, j) and a value $Q(r_{i,j-1})$ of a quantized residual sample of coordinates (i, j-1). That is, instead of encoding a value $Q(r_{i,j})$ of a quantized residual sample of coordinates (i, j), a difference value calculated by using a value $Q(r_{i,j-1})$ of a quantized residual sample of coordinates (i, j-1) as a prediction value is derived as a modified quantized residual sample value $r'_{i,j}$, and then the value $r'_{i,j}$ is encoded.

[366] As described above, the process of modifying a current quantized residual sample value by using a nearby quantized residual sample value as a prediction value may be called BDPCM prediction.

[367] Finally, the image encoding apparatus may encode a modified quantized residual block including the modified quantized residual samples and may transmit the resulting block to the image decoding apparatus. Herein, as described above, transformation of the modified quantized residual block is not performed.

Date Reçue/Date Received 2023-12-29

[368] FIG. 32 is a view showing modified quantized residual blocks generated by performing BDPCM of the present disclosure.

[369] In FIG. 32, horizontal BDPCM shows a modified quantized residual block generated according to Equation 14 when the prediction direction of BDPCM is a horizontal direction. In addition, vertical BDPCM shows a modified quantized residual block generated according to Equation 15 when the prediction direction of BDPCM is a vertical direction.

[370] FIG. 33 is a flowchart showing a procedure for encoding a current block by applying BDPCM in an image encoding apparatus.

[371] First, when a current block that is an encoding target block is input in step S3310, prediction may be performed on the current block to generate a prediction block in step S3320. The prediction block in step S3320 may be an intra-predicted block, and an intra prediction mode may be determined as described above. On the basis of the prediction block generated in step S3320, a residual block of the current block may be generated in step S3330. For example, the image encoding apparatus may generate a residual block (values of residual samples) by subtracting a prediction block (values of predicted samples) from the current block (values of original samples). For example, by performing step S3330, a residual block of FIG. 31 may be generated. On the residual block generated in step S3330, quantization may be performed in step S3340 to generate a quantized residual block, and BDPCM prediction may be performed on the quantized residual block in step S3350. The quantized residual block generated as a result of performing step S3340 may be the quantized residual block of FIG. 31. As a result of BDPCM prediction in step S3350, a modified quantized residual block of FIG. 32 may be generated according to a prediction direction. Since BDPCM prediction in step S3350 has been described with reference to FIGS. 31 to 32, a detailed description thereof will be omitted. Afterward, the image encoding apparatus may encode the modified quantized residual block in step S3360 to generate a bitstream. Herein, transformation of the modified quantized residual block may be skipped.

[372] The BDPCM operation in the image encoding apparatus described with reference to FIGS. 31 to 33 may be performed in reverse by the image decoding apparatus.

[373] FIG. 34 is a flowchart showing a procedure for reconstructing a current block by applying BDPCM in the image decoding apparatus.

[374] The image decoding apparatus may obtain information (image information) required for reconstructing the current block from a bitstream in step S3410. The information required for reconstructing the current block may include information (prediction information) on prediction of the current block, and information (residual information) on a residual of the current block. The image decoding apparatus may perform prediction on the current block on the basis of information on the current block, and may generate a prediction block in step S3420. Prediction on the current block may be intra prediction, and a detailed description thereof is the same as that described above with reference to FIG. 33. In FIG. 34, it is shown that step

Date Reçue/Date Received 2023-12-29

S3420 of generating the prediction block for the current block is performed before steps S3430 to S3450 of generating a residual block of the current block. However, no limitation thereto is imposed. After a residual block of the current block is generated, a prediction block of the current block may be generated. Alternatively, a residual block of the current block and a prediction block of the current block may be generated simultaneously.

[375] The image decoding apparatus may generate a residual block of the current block in step S3430 by parsing residual information of the current block from the bitstream. The residual block generated in step S3430 may be the modified quantized residual block shown in FIG. 32.

[376] The image decoding apparatus may generate the quantized residual block of FIG. 31 by performing BDPCM prediction in step S3440 on the modified quantized residual block of FIG. 32. BDPCM prediction in step S3440 is a procedure for generating the quantized residual block of FIG. 31 from the modified quantized residual block of FIG. 32, which corresponds to the reverse process of step S3350 performed by the image encoding apparatus. For example, when differential encoding mode information (e.g., bdpcm_flag) obtained from a bitstream indicates a differential encoding mode in which differential encoding of a residual coefficient is performed as BDPCM is applied, the image decoding apparatus performs differential encoding on a residual block to derive a modified residual block. Using a residual coefficient to be modified and a prediction residual coefficient, the image decoding apparatus may modify at least one residual coefficient to be modified among residual coefficients in a residual block. The prediction residual coefficient may be determined on the basis of the prediction direction indicated by differential encoding direction information (e.g., bdpcm_dir_flag) obtained from the bitstream. The differential encoding direction information may indicate either a vertical direction or a horizontal direction. The image decoding apparatus may assign a value obtained by adding the residual coefficient to be modified and the prediction residual coefficient, to a location of the residual coefficient to be modified. Herein, the prediction residual coefficient may be a coefficient that is immediately before and adjacent to the residual coefficient to be modified, in terms of the order according to the prediction direction.

[377] Hereinafter, BDPCM prediction in step S3440 performed by the image decoding apparatus will be described in more detail below. The decoding apparatus may calculate a quantized residual sample $Q(r_{i,j})$ by performing the calculation performed by the encoding apparatus in reverse. For example, when the prediction direction of BDPCM is a horizontal direction, the image decoding apparatus may generate a quantized residual block from a modified quantized residual block by using Equation 16.

[378] [Equation 16]

[379] $$Q(r_{i,j}) = \sum_{k=0}^{i} r'_{k,j}, \quad 0 \le i \le (M-1), \quad 0 \le j \le (N-1)$$

Date Reçue/Date Received 2023-12-29

[380] As defined in Equation 16, a value $Q(r_{i,j})$ of a quantized residual sample of coordinates (i, j) may be calculated by adding up values of modified quantized residual samples starting from coordinates (0, j) to coordinates (i, j).

[381] Alternatively, using Equation 17 instead of Equation 16, a value $Q(r_{i,j})$ of a quantized residual sample of coordinates (i, j) may be calculated.

[382] [Equation 17]

[383] $$Q(r_{i,j}) = \begin{cases} r'_{i,j}, & i = 0, \quad 0 \le j \le (N-1) \\ r'_{i,j} + Q(r_{(i-1),j}), & 1 \le i \le (M-1), \quad 0 \le j \le (N-1) \end{cases}$$

[384] Equation 17 is the reverse process corresponding to Equation 14. According to Equation 17, a value $Q(r_{0,j})$ of a quantized residual sample of coordinates (0, j) is derived as a value $r'_{0,j}$ of a modified quantized residual sample of coordinates (0, j). $Q(r_{i,j})$ of other coordinates (i, j) is derived as the sum of a value $r'_{i,j}$ of a modified quantized residual sample of coordinates (i, j) and a value $Q(r_{i-1,j})$ of a quantized residual sample of coordinates (i-1, j). That is, a difference value $r'_{i,j}$ is added up by using a value $Q(r_{i-1,j})$ of a quantized residual sample of coordinates (i-1, j) as a prediction value, thereby deriving a quantized residual sample value $Q(r_{i,j})$.

[385] When the prediction direction of BDPCM is a vertical direction, the image decoding apparatus may generate a quantized residual block from a modified quantized residual block by using Equation 18.

[386] [Equation 18]

[387] $$Q(r_{i,j}) = \sum_{k=0}^{j} r'_{i,k}, \quad 0 \le i \le (M-1), \quad 0 \le j \le (N-1)$$

[388] As defined in Equation 18, a value $Q(r_{i,j})$ of a quantized residual sample of coordinates (i, j) may be calculated by adding up values of modified quantized residual samples starting from coordinates (i, 0) to coordinates (i, j).

[389] Alternatively, using Equation 19 instead of Equation 18, a value $Q(r_{i,j})$ of a quantized residual sample of coordinates (i, j) may be calculated.

[390] [Equation 19]

[391] $$Q(r_{i,j}) = \begin{cases} r'_{i,j}, & 0 \le i \le (M-1), \quad j = 0 \\ r'_{i,j} + Q(r_{i,(j-1)}), & 0 \le i \le (M-1), \quad 1 \le j \le (N-1) \end{cases}$$

[392] Equation 19 is the reverse process corresponding to Equation 15. According to Equation 19, a value $Q(r_{i,0})$ of a quantized residual sample of coordinates (i, 0) is derived as a value $r'_{i,0}$ of a modified quantized residual sample of coordinates (i, 0). $Q(r_{i,j})$ of other coordinates (i, j) is derived as the sum of a value $r'_{i,j}$ of a modified quantized residual sample of coordinates (i, j) and a value $Q(r_{i,j-1})$ of a quantized residual sample of coordinates (i, j-1). That is, a difference value $r'_{i,j}$ is added up by using a value $Q(r_{i,j-1})$ of a quantized residual

Date Reçue/Date Received 2023-12-29

sample of coordinates (i, j-1) as a prediction value, thereby deriving a quantized residual sample value $Q(r_{i,j})$.

[393] When a quantized residual block composed of quantized residual samples is generated by performing step S3440 according to the above-described method, the image decoding apparatus performs dequantization on the quantized residual block in step S3450 to generate a residual block of the current block. When BDPCM is applied, transformation of the current block is skipped as described above. Therefore, inverse transformation of a dequantized residual block may be skipped.

[394] Afterward, the image decoding apparatus may reconstruct the current block in step S3460 on the basis of the prediction block generated in step S3420 and the residual block generated in step S3450. For example, the image decoding apparatus may reconstruct the current block (values of reconstructed samples) by adding the prediction block (values of predicted samples) and the residual block (values of residual samples). For example, a reconstructed sample value may be generated by adding a dequantized quantized sample $Q^{-1}(Q(r_{i,j}))$ to an intra block prediction value. Differential encoding mode information indicating whether BDPCM is applied to the current block may be signaled through a bitstream. In addition, when BDPCM is applied to the current block, differential encoding direction information indicating the prediction direction of BDPCM may be signaled through a bitstream. When BDPCM is not applied to the current block, the differential encoding direction information may not be signaled.

[395] FIGS. 35 to 37 are views schematically showing syntax for signaling information on BDPCM.

[396] FIG. 35 is a view showing syntax of a sequence parameter set according to an embodiment for signaling BDPCM information. In an embodiment, all SPS RBSPs included in at least one access unit (AU) having a value of 0 as a temporal ID (TemporalId) or provided through external means may be set to be used before being referenced in a decoding process. In addition, an SPS NAL unit including an SPS RBSP may be set to have nuh_layer_id that is the same as nuh_layer_id of a PPS NAL unit referring to the SPS NAL unit. In CVS, all SPS NAL units having a particular sps_seq_parameter_set_id value may be set to have the same content. In the seq_parameter_set_rbsp() syntax of FIG. 35, sps_transform_skip_enable_flag, described above, and sps_bdpcm_enabled_flag, described later, are disclosed.

[397] The syntax element sps_bdpcm_enabled_flag may indicate whether for an intra coding unit, intra_bdpcm_flag is provided in CU syntax. For example, a first value (e.g., 0) of the sps_bdpcm_enabled_flag may indicate that for an intra coding unit, the intra_bdpcm_flag is not provided in CU syntax. A second value (e.g., 1) of the sps_bdpcm_enabled_flag may indicate that for an intra coding unit, the intra_bdpcm_flag is provided in CU syntax. In the meantime, when the sps_bdpcm_enabled_flag is not provided, the value of the sps_bdpcm_enabled_flag may be set to the first value (e.g., 0).

Date Reçue/Date Received 2023-12-29

[398] FIG. 36 is a view showing an embodiment of syntax for signaling whether limitation on BDPCM is applied. In an embodiment, a predetermined limitation condition in the encoding/decoding process may be signaled using general_constraint_info() syntax. Using the syntax of FIG. 36, the syntax element no_bdpcm_constraint_flag indicating whether the value of the sps_bdpcm_enabled_flag described above needs to be set to 0 may be signaled. For example, a first value (e.g., 0) of the no_bdpcm_constraint_flag may indicate that such a limitation is not applied. When the value of the no_bdpcm_constraint_flag is a second value (e.g., 1), the value of the sps_bdpcm_enabled_flag may be forced to be a first value (e.g., 0).

[399] FIG. 37 is a view showing an embodiment of coding unit() syntax for signaling information on BDPCM for an encoding unit. As shown in FIG. 37, the syntax elements intra_bdpcm_flag and intra_bdpcm_dir_flag may be signaled using coding_unit() syntax.The syntax element intra_bdpcm_flag may indicate whether BDPCM is applied to a current luma encoding block located at (x0, y0).

[400] For example, a first value (e.g., 0) of the intra_bdpcm_flag may indicate that BDPCM is not applied to the current luma encoding block. A second value (e.g., 1) of the intra_bdpcm_flag may indicate that BDPCM is applied to the current luma encoding block. By indicating that BDPCM is applied, the intra_bdpcm_flag may indicate whether transformation is skipped and also whether an intra luma prediction mode is performed by the intra_bdpcm_dir_flag, which will be described later.

[401] In the meantime, the value of the above-described variable BdpcmFlag[x][y] may be set to the value of the intra_bdpcm_flag, with respect to x = x0..x0 + cbWidth - 1 and y = y0..y0 + cbHeight – 1.

[402] The syntax element intra_bdpcm_dir_flag may indicate a prediction direction of BDPCM. For example, a first value (e.g., 0) of the intra_bdpcm_dir_flag may indicate that the BDPCM prediction direction is a horizontal direction. A second value (e.g., 1) of intra_bdpcm_dir_flag may indicate that the BDPCM prediction direction is a vertical direction.

[403] In the meantime, the value of the variable BdpcmDir[x][y] may be set to the value of the intra_bdpcm_dir_flag, with respect to x = x0 .. x0 + cbWidth - 1 and y = y0 .. y0 + cbHeight – 1.

[404] <u>Intra prediction on chroma block</u>

[405] When intra prediction is performed on a current block, prediction on a luma component block (luma block) of the current block and prediction on a chroma component block (chroma block) may be performed. In this case, an intra prediction mode for the chroma block may be set separately from an intra prediction mode for the luma block.

[406] For example, the intra prediction mode for the chroma block may be indicated on the basis of intra chroma prediction mode information. The intra chroma prediction mode information may be signaled in the form of a syntax element intra_chroma_pred_mode. For example, the intra chroma prediction mode information may indicate one of the following: a

Date Reçue/Date Received 2023-12-29

planar mode, a DC mode, a vertical mode, a horizontal mode, a derived mode (DM), and a cross-component linear model (CCLM) mode. Herein, the planar mode may refer to intra prediction mode 0, the DC mode may refer to intra prediction mode 1, the vertical mode may refer to intra prediction mode 26, and the horizontal mode may refer to intra prediction mode 10. The DM may also be referred to as a direct mode. The CCLM may also be referred to as a linear model (LM). The CCLM mode may include any one of the following: L_CCLM, T_CCLM, and LT_CCLM.

[407] In the meantime, the DM and the CCLM are dependent intra prediction modes for predicting a chroma block by using information of a luma block. The DM may refer to a mode in which the same intra prediction mode as the intra prediction mode for the luma component is applied as the intra prediction mode for the chroma component. In addition, the CCLM may refer to an intra prediction mode in which in a process of generating a prediction block for a chroma block, reconstructed samples of a luma block are sub-sampled, and then samples derived by applying CCLM parameters α and β to the sub-sampled samples are used as prediction samples of the chroma block.

[408] Overview of cross-component linear model (CCLM) mode

[409] As described above, the CCLM mode may be applied to a chroma block. The CCLM mode is the intra prediction mode in which the correlation between a luma block and a chroma block corresponding to the luma block is used, and is performed by deriving a linear model on the basis of neighboring samples of the luma block and neighboring samples of the chroma block. In addition, a prediction sample of the chroma block may be derived on the basis of the derived linear model and the reconstructed samples of the luma block.

[410] Specifically, when the CCLM mode is applied to a current chroma block, parameters for a linear model may be derived on the basis of neighboring samples used for intra prediction of the current chroma block and neighboring samples used for intra prediction of a current luma block. For example, a linear model for a CCLM may be represented on the basis of the equation as below.

[411] [Equation 20]

[412] $$pred_c(i,j)=\alpha \cdot rec_L'(i,j)+\beta$$

[413] Herein, $pred_c(i,j)$ may refer to a prediction sample of coordinates (i,j) of the current chroma block in the current CU. $rec_L'(i,j)$ may refer to a reconstructed sample of coordinates (i,j) of the current luma block in the CU. For example, the $rec_L'(i,j)$ may refer to a down-sampled reconstructed sample of the current luma block. Linear model coefficients α and β may be signaled, or may be derived from a neighboring sample.

[414] Joint coding of residuals (joint CbCr)

[415] In an encoding/decoding process according to an embodiment, chroma residuals may be encoded/decoded together. This may be referred to as joint coding of residuals or as joint

Date Reçue/Date Received 2023-12-29

CbCr. Whether to apply (activate) a joint coding mode of CbCr may be signaled by a joint coding mode signaling flag tu_joint_cbcr_residual_flag that is signaled at the level of the transformation basis. In addition, a selected encoding mode may be derived by chroma CBFs. The flag tu_joint_cbcr_residual_flag may be present when a value of at least one chroma CBF for the transformation basis is 1. A chroma QP offset value indicates a difference between a general chroma QP offset value signaled for a regular chroma residual encoding mode and a chroma QP offset value for a CbCr joint coding mode. The chroma QP offset value may be signaled through a PPS or a slice header. This QP offset value may be used to drive a chroma QP value for blocks using a joint chroma residual encoding mode.

[416] In the case in which Mode 2, which is a corresponding joint chroma encoding mode, in the table below is activated for the transformation basis, while quantization and decoding of the transformation basis are performed, a chroma QP offset thereof may be added to a target luma-derived chroma QP (applied luma-derived chroma QP).

[417] Regarding other modes like Modes 1 and 3 in the table below, a chroma QP may be derived in such a manner that it is obtained for a general Cb or Cr block. Such a process of reconstructing chroma residuals (resCb and resCr) from a transformation block may be selected according to the table below. When the present mode is activated, one single joint chroma residual block (resJointC[x][y] in the table below) is signaled, and a residual block resCb for a Cb and a residual block resCr for a Cr may be derived considering information such as tu_cbf_cb, tu_cbf_cr, and CSign that is a sign value disclosed in a slice header.

[418] In the encoding apparatus, a joint chroma component may be derived as follows. According to a joint coding mode, resJointC{1, 2} may be generated according to the following order. In the case of Mode 2 (single residual with reconstruction Cb = C, Cr = CSign * C), a joint residual may be determined according to the equation below.

[419] [Equation 21]

[420] resJointC[x][y] = (resCb[x][y] + CSign * resCr[x][y]) / 2.

[421] Alternatively, in the case of Mode 1 (single residual with reconstruction Cb = C, Cr = (CSign * C) / 2), a joint residual may be determined according to the equation below.

[422] [Equation 22]

[423] resJointC[x][y] = (4 * resCb[x][y] + 2 * CSign * resCr[x][y]) / 5.

[424] Alternatively, in the case of Mode 3 (single residual with reconstruction Cr = C, Cb = (CSign * C) / 2), a joint residual may be determined according to the equation below.

[425] [Equation 23]

[426] resJointC[x][y] = (4 * resCr[x][y] + 2 * CSign * resCb[x][y]) / 5.

[427] [Table 2]

tu_cbf_cb	tu_cbf_cr	reconstruction of Cb and Cr residuals	mode

Date Reçue/Date Received 2023-12-29

1	0	resCb[x][y] = resJointC[x][y] resCr[x][y] = (CSign * resJointC[x][y]) >> 1	1
1	1	resCb[x][y] = resJointC[x][y] resCr[x][y] = CSign * resJointC[x][y]	2
0	1	resCb[x][y] = (CSign * resJointC[x][y]) >> 1 resCr[x][y] = resJointC[x][y]	3

[428] The above table shows reconstruction of chroma residuals. CSign refers to a sign value +1 or –1 specified in a slice header. resJointC[ ][ ] refers to a transmitted residual. In the table, the modes refer to TuCResMode, which will be described later. The three joint chroma encoding modes in the table may be supported only for an I slice. For P and B slices, only mode 2 may be supported. Therefore, for P and B slices, the syntax element tu_joint_cbcr_residual_flag may be provided only when both chroma cbf (e.g., tu_cbf_cb and tu_cbf_cr) values are 1. In the meantime, in context modeling of tu_cbf_luma and tu_cbf_cb, a transformation depth may be removed.

[429] Embodiment 1 : QP update method using ACT Qp_offset

[430] As described above, the update of the QP to apply ACT may be performed. The above-described update of the QP has several problems. For example, when the above-described method is used, it is impossible to set different ACT Qp offsets for individual color components. Further, the derived qP value may have a negative value. Accordingly, in the following embodiment, described is a method of applying clipping to a Qp value derived on the basis of an ACT QP offset value of a color component value.

[431] In an embodiment, a quantization parameter qP may be derived as follows.

[432] First, when the cIdx has a value of 0, the qP and the ACT Qp offset may be derived as shown in the equation below.

[433] [Equation 24]

[434] $qP = Qp'_{Y}$

[435] ActQpOffset = 5

[436] Alternatively, when the TuCResMode[xTbY][yTbY] has a value of 2, the qP and the ACT Qp offset may be derived as shown in the equation below.

[437] [Equation 25]

[438] $qP = Qp'_{CbCr}$

[439] ActQpOffset =5

[440] Alternatively, when the cIdx has a value of 1, the qP and the ACT Qp offset may be derived as shown in the equation below.

[441] [Equation 26]

[442] $qP = Qp'_{Cb}$

[443] ActQpOffset = 3

Date Reçue/Date Received 2023-12-29

[444] The quantization parameter qP may be updated as follows.

[445] When the transform_skip_flag[xTbY][yTbY][cIdx] has a value of 0, the qP may be derived as shown in the equation below.

[446] [Equation 27]

[447] qP = Max(0, qP - (cu_act_enabled_flag[xTbY][yTbY] ? ActQpOffset : 0))

[448] Alternatively, when the transform_skip_flag[xTbY][yTbY][cIdx] has a value of 1, the qP may be derived as shown in the equation below.

[449] [Equation 28]

[450] qP = Max(0, Max(QpPrimeTsMin, qP) - (cu_act_enabled_flag[xTbY][yTbY] ? ActQpOffset : 0))

[451] In another embodiment, when the transform_skip_flag[xTbY][yTbY][cIdx] has a value of 1, the qP may be clipped using a value of the QpPrimeTsMin instead of 0 as shown in the equation below.

[452] [Equation 29]

[453] qP = Max(QpPrimeTsMin, qP - (cu_act_enabled_flag[xTbY][yTbY] ? ActQpOffset : 0)

[454] In the meantime, in another embodiment, a quantization parameter qP may be derived as follows.

[455] First, when the cIdx has a value of 0, the qP and the ACT Qp offset may be derived as shown in the equation below.

[456] [Equation 30]

[457] $qP = Qp'_{Y}$

[458] ActQpOffset = 5

[459] Alternatively, when the TuCResMode[xTbY][yTbY] has a value of 2, the qP and the ACT Qp offset may be derived as shown in the equation below.

[460] [Equation 31]

[461] $qP = Qp'_{CbCr}$

[462] ActQpOffset =5

[463] Alternatively, when the cIdx has a value of 1, the qP and the ACT Qp offset may be derived as shown in the equation below.

[464] [Equation 32]

[465] $qP = Qp'_{Cb}$

[466] ActQpOffset = 5

[467] Alternatively, when the cIdx has a value of 2, the qP and the ACT Qp offset may be derived as shown in the equation below.

[468] [Equation 33]

[469] $qP = Qp'_{Cr}$

[470] ActQpOffset = 3

Date Reçue/Date Received 2023-12-29

[471] The quantization parameter qP may be updated as follows.

[472] When the transform_skip_flag[xTbY][yTbY][cIdx] has a value of 0, the qP may be derived as shown in the equation below.

[473] [Equation 34]

[474] qP = Max(0, qP - (cu_act_enabled_flag[xTbY][yTbY] ? ActQpOffset : 0))

[475] Alternatively, when the transform_skip_flag[xTbY][yTbY][cIdx] has a value of 1, the qP may be derived as shown in the equation below.

[476] [Equation 35]

[477] qP = Max(0, Max(QpPrimeTsMin, qP) - (cu_act_enabled_flag[xTbY][yTbY] ? ActQpOffset : 0))

[478] In another embodiment, when the transform_skip_flag[xTbY][yTbY][cIdx] has a value of 1, the qP may be clipped using a value of the QpPrimeTsMin instead of 0 as shown in the equation below.

[479] [Equation 36]

[480] qP = Max(QpPrimeTsMin, qP - (cu_act_enabled_flag[xTbY][yTbY] ? ActQpOffset : 0))

[481] In the meantime, in still another embodiment, a quantization parameter qP may be derived as follows.

[482] First, when the cIdx has a value of 0, the qP and the ACT Qp offset may be derived as shown in the equation below.

[483] [Equation 37]

[484] $qP = Qp'_{Y}$

[485] ActQpOffset = -5

[486] Alternatively, when the TuCResMode[xTbY][yTbY] has a value of 2, the qP and the ACT Qp offset may be derived as shown in the equation below.

[487] [Equation 38]

[488] $qP = Qp'_{CbCr}$

[489] ActQpOffset =-5

[490] Alternatively, when the cIdx has a value of 1, the qP and the ACT Qp offset may be derived as shown in the equation below.

[491] [Equation 39]

[492] $qP = Qp'_{Cb}$

[493] ActQpOffset = -5

[494] Alternatively, when the cIdx has a value of 2, the qP and the ACT Qp offset may be derived as shown in the equation below.

[495] [Equation 40]

[496] $qP = Qp'_{Cr}$

[497] ActQpOffset = -3

Date Reçue/Date Received 2023-12-29

[498] The quantization parameter qP may be updated as follows.

[499] When the transform_skip_flag[xTbY][yTbY][cIdx] has a value of 0, the qP may be derived as shown in the equation below.

[500] [Equation 41]

[501] qP = Max(0, qP + (cu_act_enabled_flag[xTbY][yTbY] ? ActQpOffset : 0))

[502] Alternatively, when the transform_skip_flag[xTbY][yTbY][cIdx] has a value of 1, the qP may be derived as shown in the equation below.

[503] [Equation 42]

[504] qP = Max(0, Max(QpPrimeTsMin, qP) + (cu_act_enabled_flag[xTbY][yTbY] ? ActQpOffset : 0))

[505] In the meantime, in another embodiment, when the transform_skip_flag[xTbY][yTbY][cIdx] has a value of 1, the qP may be clipped using a value of the QpPrimeTsMin instead of 0 as shown in the equation below.

[506] [Equation 43]

[507] qP = Max(QpPrimeTsMin, qP + (cu_act_enabled_flag[xTbY][yTbY] ? ActQpOffset : 0))

[508] In the above description, Y, Cb, and Cr may denote three color components. For example, in ACT transformation, Y may correspond to C0. Cb may correspond to C1 or Cg. In addition, Cr may correspond to C2 or Co.

[509] In addition, the ACTQpOffset values of -5, -5, and -3 for the three color components may be replaced with other values or other variables.

[510] <u>Embodiment 2 : signaling of QP offset adjustment for ACT</u>

[511] In the above-described embodiment, ACT QP offset adjustments are fixed to -5, -5, and -3 for Y, Cg, and Co components. In the present embodiment, in order to provide more flexibility to an ACT QP adjustment offset, a method of signaling an ACT QP offset will be described. The ACT QP offset may be signaled as a parameter in a PPS.

[512] In an embodiment, qp_offset may be signaled according to a syntax table of FIG. 38. The syntax elements therefor are as follows.

[513] A syntax element pps_act_qp_offsets_present_flag may indicate whether a syntax element related to an ACT QP offset is present in a PPS. For example, the pps_act_qp_offsets_present_flag may indicate whether syntax elements pps_act_y_qp_offset, pps_act_cb_qp_offset, and pps_act_cr_qp_offset, which will be described later, are signaled as a PPS.

[514] For example, a first value (e.g., 0) of the pps_act_qp_offsets_present_flag may indicate that the pps_act_y_qp_offset, the pps_act_cb_qp_offset, and the pps_act_cr_qp_offset are not signaled through a PPS syntax table.

Date Reçue/Date Received 2023-12-29

[515] A second value (e.g., 1) of the pps_act_qp_offsets_present_flag may indicate that the pps_act_y_qp_offset, the pps_act_cb_qp_offset, and the pps_act_cr_qp_offset are signaled through a PPS syntax table.

[516] When the pps_act_qp_offsets_present_flag is not provided from a bitstream, the pps_act_qp_offsets_present_flag may be derived as the first value (e.g., 0). For example, when a flag (e.g., sps_act_enabled_flag signaled in an SPS) indicating whether ACT is applied has a first value (e.g., 0) indicating that ACT is not applied, the pps_act_qp_offsets_present_flag may be forced to have a first value (e.g., 0).

[517] When a value of a syntax element cu_act_enabled_flag is a second value (e.g., 1) indicating that ACT is applied for the current encoding basis, syntax elements pps_act_y_qp_offset_plus5, pps_act_cb_qp_offset_plus5s, and pps_act_cr_qp_offset_plus3 may be used to determine offsets applied to quantization parameter values qP for luma, Cb, and Cr components, respectively. When the values of the pps_act_y_qp_offset_plus5, the pps_act_cb_qp_offset_plus5, and the pps_act_cr_qp_offset_plus3 are not present in a bitstream, each value may be set to 0.

[518] According to the syntax elements, a value of a variable PpsActQpOffsetY may be determined to be pps_act_y_qp_offset_plus5 – 5. A value of a variable PpsActQpOffsetCb may be determined to be pps_act_cb_qp_offset_plus5 - 5. In addition, a value of a variable PpsActQpOffsetCr may be determined to be pps_act_cb_qp_offset_plus3 – 3.

[519] Herein, ACT is not orthonormal transformation, so 5, 5, and 3 may be applied as the constant offset values to be subtracted. In an embodiment, for bitstream matching, the values of the PpsActQpOffsetY, the PpsActQpOffsetCb, and the PpsActQpOffsetCr may have values ranging from -12 to 12. In addition, according to an embodiment, in addition to 5, 5, and 3, the Qp offset values may be replaced with other constant values and used.

[520] In another embodiment, a QP may be adjusted using a more flexible ACT_QP offset. In the following embodiment, described is an example in which an ACT QP offset is signaled in a bitstream. Accordingly, the ACT QP offset may have a wider offset range. Therefore, the QP updated using the ACT QP offset is more likely to be out of an available range, so it is necessary to perform clipping on the upper and lower limits for the updated QP (more detailed embodiments will be described later in Embodiments 6 and 7)

[521] Variables PpsActQpOffsetY, PpsActQpOffsetCb, PpsActQpOffsetCr, and PpsActQpOffsetCbCr indicating ACT QP offsets may be values derived using ACT QP offsets signaled through a bitstream, or preset constants. For bitstream conformance, the PpsActQpOffsetY, the PpsActQpOffsetCb, the PpsActQpOffsetCr, and the PpsActQpOffsetCbCr may have values ranging from -12 to +12.

[522] When without using a fixed value, a value of a QP offset is signaled and its value has a value ranging from -12 to 12, it is necessary to clip the upper limit value of the derived QP

Date Reçue/Date Received 2023-12-29

value, in addition to clipping the lower limit value of the QP value derived to avoid an QP having a negative value.

[523] To prevent the value of the qP from having a negative value, the lowest value of the qP may be forced to 0. Alternatively, the lowest value of the qP may be set to the value determined by the signaled syntax element. For example, to signal the lowest value of the qP when a transformation skip mode is applied, a syntax element QpPrimeTsMin indicating a value of the qP applied when the transformation skip mode is applied may be used. The maximum value of the qP may be limited to the available maximum value (e.g., 63) of the qP or the maximum available qP value determined according to a signaled syntax element.

[524] In an embodiment according to the above, a quantization parameter qP may be derived as follows. First, when the cIdx has a value of 0, the qP and the ACT Qp offset may be derived as shown in the equation below.

[525] [Equation 44]

[526] $qP = Qp'_{Y}$

[527] ActQpOffset = PpsActQpOffsetY

[528] Alternatively, when the TuCResMode[xTbY][yTbY] has a value of 2, the qP and the ACT Qp offset may be derived as shown in the equation below.

[529] [Equation 45]

[530] $qP = Qp'_{CbCr}$

[531] ActQpOffset = PpsActQpOffsetCbCr

[532] Alternatively, when the cIdx has a value of 1, the qP and the ACT Qp offset may be derived as shown in the equation below.

[533] [Equation 46]

[534] $qP = Qp'_{Cb}$

[535] ActQpOffset = PpsActQpOffsetCb

[536] Alternatively, when the cIdx has a value of 2, the qP and the ACT Qp offset may be derived as shown in the equation below.

[537] [Equation 47]

[538] $qP = Qp'_{Cr}$

[539] ActQpOffset = PpsActQpOffsetCr

[540] In an embodiment, the quantization parameter qP may be updated as follows. When the transform_skip_flag[xTbY][yTbY][cIdx] has a value of 0, the qP may be derived as shown in the equation below.

[541] [Equation 48]

[542] qP = Clip3(0, 63, qP - (cu_act_enabled_flag[xTbY][yTbY] ? ActQpOffset : 0))

[543] Alternatively, when the transform_skip_flag[xTbY][yTbY][cIdx] has a value of 1, the qP may be derived as shown in the equation below.

[544] [Equation 49]

Date Reçue/Date Received 2023-12-29

[545] qP = Clip3(0, 63, Max(QpPrimeTsMin, qP) - (cu_act_enabled_flag[xTbY][yTbY] ? ActQpOffset : 0)

[546] In another embodiment, when the transform_skip_flag[xTbY][yTbY][cIdx] has a value of 1, the lowest value of the qP may be clipped using a value of the QpPrimeTsMin instead of 0 as shown in the equation below.

[547] [Equation 50]

[548] The quantization parameter qP may be updated as follows.

[549] When the transform_skip_flag[xTbY][yTbY][cIdx] has a value of 0, the qP may be derived as shown in the equation below.

[550] qP = Clip3(0, 63, qP - (cu_act_enabled_flag[xTbY][yTbY] ? ActQpOffset : 0))

[551] Alternatively, when the transform_skip_flag[xTbY][yTbY][cIdx] has a value of 1, the qP may be derived as shown in the equation below.

[552] qP = Clip3(QpPrimeTsMin, 63, qP - cu_act_enabled_flag[xTbY][yTbY] ? ActQpOffset : 0)

[553] In another embodiment, the quantization parameter qP may be updated as follows.

[554] When the transform_skip_flag[xTbY][yTbY][cIdx] has a value of 0, the qP may be derived as shown in the equation below.

[555] [Equation 51]

[556] qP = Clip3(0, 63+ QpBdOffset, qP + (cu_act_enabled_flag[xTbY][yTbY] ? ActQpOffset : 0))

[557] Alternatively, when the transform_skip_flag[xTbY][yTbY][cIdx] has a value of 1, the qP may be derived as shown in the equation below.

[558] [Equation 52]

[559] qP = Clip3(0, 63+ QpBdOffset, Max(QpPrimeTsMin, qP) + (cu_act_enabled_flag[xTbY][yTbY] ? ActQpOffset : 0)

[560] In another embodiment, when the transform_skip_flag[xTbY][yTbY][cIdx] has a value of 1, the lowest value of the qP may be clipped using a value of the QpPrimeTsMin instead of 0 as shown in the equation below.

[561] [Equation 53]

[562] The quantization parameter qP may be updated as follows.

[563] When the transform_skip_flag[xTbY][yTbY][cIdx] has a value of 0, the qP may be derived as shown in the equation below.

[564] qP = Clip3(0, 63+ QpBdOffset, qP + (cu_act_enabled_flag[xTbY][yTbY] ? ActQpOffset : 0))

[565] Alternatively, when the transform_skip_flag[xTbY][yTbY][cIdx] has a value of 1, the qP may be derived as shown in the equation below.

[566] qP = Clip3(QpPrimeTsMin, 63+ QpBdOffset, qP + cu_act_enabled_flag[xTbY][yTbY] ? ActQpOffset : 0)

Date Reçue/Date Received 2023-12-29

[567] Embodiment 3 : method to allow ACT when chroma BDPCM is performed

[568] In an embodiment, when BDPCM is applied to a luma component block, ACT may be applied to encode/decode the block. However, when BDPCM is applied to a chroma component block, ACT is limited not to be applied to encode/decode the block.

[569] In the meantime, even when BDPCM is applied to a chroma component block, ACT is applied to the block, thereby improving encoding efficiency. FIG. 39 shows an embodiment of a syntax configuration for applying ACT even when BDPCM is applied to a chroma component block. As shown in FIG. 39, by removing the condition for obtaining a BDPCM syntax element for a chroma component according to the value of the cu_act_enabled_flag indicating whether ACT is applied to the current encoding basis, the BDPCM syntax element therefor may be obtained regardless of whether ACT is applied to a chroma component block, and BDCPM encoding may be performed accordingly.

[570] Embodiment 4 : method of applying ACT even when encoding/decoding is performed with CCLM

[571] Both CCLM and ACT are intended to remove unnecessary overlap between components. There are some overlapping parts between CCLM and ACT, but even after applying all of these, it is impossible to completely remove overlap between components. Therefore, overlap between components may be more removed by applying CCLM and ACT together.

[572] The following embodiment describes an embodiment in which CCLM and ACT are applied together. In performing decoding, the decoding apparatus may apply CCLM first, and apply ACT. When ACT is applied to both BDPCM and CCLM for a chroma component, a syntax table for signaling this may be modified as shown in FIG. 40. Accordingly, as shown in the syntax table of FIG. 40, among the limitations for signaling a syntax element related to intra_bdpcm_chroma and cclm, if(!cu_act_enabled_flag) for signaling a syntax element depending on whether ACT is not applied may be removed from the syntax table.

[573] Embodiment 5 : method of applying flexible ACT Qp including joint CbCr

[574] When an ACT mode is applied, a prediction residual may be transformed from one color space (e.g., GBR or YCbCr) into a YCgCo color space. In addition, residuals of the transformation basis may be encoded in the YCgCo color space. As an embodiment of ACT core transformation (transformation kernel) used for color space transformation, the following transformation kernel as described above may be used.

[575] [Equation 54]

[576] $$\begin{bmatrix} C_0' \\ C_1' \\ C_2' \end{bmatrix} = \begin{bmatrix} 2 & 1 & 1 \\ 2 & -1 & -1 \\ 0 & -2 & 2 \end{bmatrix} \begin{bmatrix} C_0 \\ C_1 \\ C_2 \end{bmatrix} /4$$

[577] [Equation 55]

Date Reçue/Date Received 2023-12-29

[578] $$\begin{bmatrix} C_0 \\ C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & -1 & -1 \\ 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} C_0' \\ C_1' \\ C_2' \end{bmatrix}$$

[579] As described in the equations above, C0', C1', and C2' (herein, C0'=Y, C1'= Cg, C2'=Co) transformations are not normalized. For example, the L2 norm does not have a value of 1. For example, the L2 norm of transformation for an individual component may have a value of about 0.6 for C0' and C1', and may have a value of about 0.7 for C2'. Herein, the L2 norm is a value obtained as the square root of the sum of the squares of respective coefficients. For example, C0' = 2/4*C0 + 1/4*C1 + 1/4*C2 may be calculated. Therefore, the norm of C0' may be calculated as the square root of (2/4*2/4 + 1/4*1/4 + 1/4*1/4). Therefore, this may be calculated as the square root of 6/16, and may be calculated as having a value of about 0.6.

[580] When normalized transformation is not applied, the dynamic range of the individual component is irregular. In addition, this causes a decrease in encoding performance of a general video compression system.

[581] In order to compensate for the dynamic range of a residual signal, a QP offset value for compensating a dynamic range change for an individual transformation component is transmitted, so that QP adjustment may be performed. For example, this embodiment may be applied to a general QP adjustment control method for ACT transformation as well as joint CbCr.

[582] Individual color components are not encoded independently, but together, so that the method described above in Embodiment 3 for joint CbCr may cause a dynamic range change between the individual color components.

[583] In encoding and decoding methods according to an embodiment, ACT QP offset adjustment may be fixed to -5, which may be equally applied to Y, Cg, and Co.

[584] In an embodiment, in order to provide flexible Qp control for individual components and jointCbCr, it may be allowed to use different ACT Qp offsets for Y, Cb, Cr, and/or joint CbCr. The ACT Qp offset values may be determined on the basis of a component index, and/or joint CbCr, and/or a joint CbCr mode.

[585] In order to indicate the ACT Qp offsets, ppsActQpOffsetY, ppsActQpOffsetCb, and ppsActQpOffsetCr may be used. In addition, ppsActQpOffsetCbCr may be used for the ACT QP offset of joint CbCr mode 2 having CBF in which all Cb and Cr components have non-zero values. These values (e.g., ppsActQpOffsetY, ppsActQpOffsetCb, ppsActQpOffsetCr, and ppsActQpOffsetCbCr) may be pre-determined to be predetermined values, or signaled through a bitstream. The ACT QP offset of the joint CbCr mode may be set in another method or to another value.

[586] In an embodiment, the ACT Qp offsets of -5, -5, and -3 may be used for Y, Cb, and Cr, and the ACT Qp offset of -4 may be used for joint CbCr.

Date Reçue/Date Received 2023-12-29

[587] In another embodiment, the ACT Qp offsets of -5, -4, and -3 may be used for Y, Cb, and Cr, and the ACT Qp offset of -3 may be used for the joint CbCr mode in which the value of tu_cbf_cb is not 0.

[588] In still another embodiment, the ACT QP offset of joint CbCr mode 2 may have its own offset value. For another joint CbCr mode, the ACT QP offset may use the offset of a corresponding component. For example, a quantization parameter qP may be determined as follows. First, when the cIdx has a value of 0, the qP and the ACT Qp offset may be derived as shown in the equation below.

[589] [Equation 56]

[590] $qP = Qp'_{Y}$

[591] ActQpOffset = ppsActQpOffsetY

[592] Alternatively, when the TuCResMode[xTbY][yTbY] has a value of 2, the qP and the ACT Qp offset may be derived as shown in the equation below.

[593] [Equation 57]

[594] $qP = Qp'_{CbCr}$

[595] ActQpOffset = ppsActQpOffsetCbCr

[596] Alternatively, when the cIdx has a value of 1, the qP and the ACT Qp offset may be derived as shown in the equation below.

[597] [Equation 58]

[598] $qP = Qp'_{Cb}$

[599] ActQpOffset = ppsActQpOffsetCb

[600] Alternatively, when the cIdx has a value of 2, the qP and the ACT Qp offset may be derived as shown in the equation below.

[601] [Equation 59]

[602] $qP = Qp'_{Cr}$

[603] ActQpOffset = ppsActQpOffsetCr

[604] In an embodiment, the quantization parameter qP may be updated as follows.

[605] When the transform_skip_flag[xTbY][yTbY][cIdx] has a value of 0, the qP may be derived as shown in the equation below.

[606] [Equation 60]

[607] qP = Clip3(0, 63+ QpBdOffset, qP + (cu_act_enabled_flag[xTbY][yTbY] ? ActQpOffset : 0))

[608] Alternatively, when the transform_skip_flag[xTbY][yTbY][cIdx] has a value of 1, the qP may be derived as shown in the equation below.

[609] [Equation 61]

[610] qP = Clip3(QpPrimeTsMin, 63+ QpBdOffset, qP + cu_act_enabled_flag[xTbY][yTbY] ? ActQpOffset : 0)

Date Reçue/Date Received 2023-12-29

[611] In another embodiment, with respect to a joint CbCr mode (e.g., in the case of Modes 1 and 2) with tu_cbf_cb!=0, an offset for joint CbCr may be determined using the ppsActQpOffsetCb. Alternatively, with respect to a joint CbCr mode (e.g., in the case of Mode 3) with tu_cbf_cb==0, an offset for joint CbCr may be determined using the ppsActQpOffsetCr. For example, the above-described embodiment may be modified and applied as follows.

[612] The quantization parameter qP may be updated as follows. First, when the cIdx has a value of 0, the qP and the ACT Qp offset may be derived as shown in the equation below.

[613] [Equation 62]

[614] $qP = Qp'_{Y}$

[615] ActQpOffset = ppsActQpOffsetY

[616] Alternatively, when the TuCResMode[xTbY][yTbY] has a value of 2, the qP may be derived as shown in the equation below.

[617] [Equation 63]

[618] $qP = Qp'_{CbCr}$

[619] Alternatively, when the cIdx has a value of 1, the qP and the ACT Qp offset may be derived as shown in the equation below.

[620] [Equation 64]

[621] $qP = Qp'_{Cb}$

[622] ActQpOffset = ppsActQpOffsetCb

[623] Alternatively, when the cIdx has a value of 2, the qP and the ACT Qp offset may be derived as shown in the equation below.

[624] [Equation 65]

[625] $qP = Qp'_{Cr}$

[626] ActQpOffset = ppsActQpOffsetCr

[627] When the cIdx does not have a value of 0 and the TuCResMode[xTbY][yTbY] does not have a value of 0, the ACT Qp offset for the joint CbCr mode may be determined according to the pseudocode below.

[628] [Equation 66]

[629] if (TuCResMode[xTbY][yTbY] is euqal to 1 or 2)

[630] ActQpOffset= ppsActQpOffsetCb;

[631] else

[632] ActQpOffset= ppsActQpOffsetCr;

[633] In an embodiment, the quantization parameter qP may be updated as follows. When the transform_skip_flag[xTbY][yTbY][cIdx] has a value of 0, the qP may be derived as shown in the equation below.

[634] [Equation 67]

[635] qP = Clip3(0, 63+ QpBdOffset, qP + (cu_act_enabled_flag[xTbY][yTbY] ? ActQpOffset : 0))

Date Reçue/Date Received 2023-12-29

[636] Alternatively, when the transform_skip_flag[xTbY][yTbY][cIdx] has a value of 1, the qP may be derived as shown in the equation below.

[637] [Equation 68]

[638] qP = Clip3(QpPrimeTsMin, 63+ QpBdOffset, qP + cu_act_enabled_flag[xTbY][yTbY] ? ActQpOffset : 0)

[639] In another embodiment, regardless of the joint CbCr mode, ppsActQpOffsetY is used when the component index is Y, ppsActQpOffsetCb is used when the component index is Cb, and ppsActQpOffsetCr is used when the component index is Cr, whereby the qP may be derived. For example, the quantization parameter qP may be derived as follows.

[640] First, when the cIdx has a value of 0, the qP and the ACT Qp offset may be derived as shown in the equation below.

[641] [Equation 69]

[642] $qP = Qp'_{Y}$

[643] ActQpOffset = ppsActQpOffsetY

[644] Alternatively, when the TuCResMode[xTbY][yTbY] has a value of 2, the qP and the ACT Qp offset may be derived as shown in the equation below.

[645] [Equation 70]

[646] $qP = Qp'_{CbCr}$

[647] ActQpOffset =(cIdx==1)? ppsActQpOffsetCb: ppsActQpOffsetCr

[648] Alternatively, when the cIdx has a value of 1, the qP and the ACT Qp offset may be derived as shown in the equation below.

[649] [Equation 71]

[650] $qP = Qp'_{Cb}$

[651] ActQpOffset = ppsActQpOffsetCb

[652] Alternatively, when the cIdx has a value of 2, the qP and the ACT Qp offset may be derived as shown in the equation below.

[653] [Equation 72]

[654] $qP = Qp'_{Cr}$

[655] ActQpOffset = ppsActQpOffsetCr

[656] The quantization parameter qP may be updated as follows.

[657] When the transform_skip_flag[xTbY][yTbY][cIdx] has a value of 0, the qP may be derived as shown in the equation below.

[658] [Equation 73]

[659] qP = Clip3(0, 63+ QpBdOffset, qP + (cu_act_enabled_flag[xTbY][yTbY] ? ActQpOffset : 0))

[660] Alternatively, when the transform_skip_flag[xTbY][yTbY][cIdx] has a value of 1, the qP may be derived as shown in the equation below.

[661] [Equation 74]

Date Reçue/Date Received 2023-12-29

[662] qP = Clip3(QpPrimeTsMin, 63+ QpBdOffset, qP + cu_act_enabled_flag[xTbY][yTbY] ? ActQpOffset : 0)

[663] Embodiment 6 : method of signaling ACT Qp offset including joint CbCr

[664] Hereinafter, described is an example in which an ACT QP offset is signaled through a bitstream to provide more flexibility. ACT QP offsets may be signaled through a SPS, a PPS, a picture header, a slice header, or header sets of other types. An ACT Qp offset of joint CbCr may be signaled separately, or may be derived from ACT QP offsets for Y, Cb, and Cr.

[665] Without loss of generality, FIG. 41 shows an example of a syntax table in which an ACT Qp offset is signaled in a PPS. As in the embodiment of FIG. 41, one ACT Qp offset may be signaled for joint CbCr. The syntax elements indicated in the syntax table of FIG. 41 will be described.

[666] A syntax element pps_act_qp_offsets_present_flag may indicate whether a syntax element related to an ACT QP offset is present in a PPS. For example, the pps_act_qp_offsets_present_flag may indicate whether syntax elements pps_act_y_qp_offset_plusX1, pps_act_cb_qp_offset_plusX2, pps_act_cr_qp_offset_plusX3, and pps_act_cbcr_qp_offset_plusX4, which will be described later, are signaled as a PPS.

[667] For example, a first value (e.g., 0) of the pps_act_qp_offsets_present_flag may indicate that the pps_act_y_qp_offset_plusX1, the pps_act_cb_qp_offset_plusX2, the pps_act_cr_qp_offset_plusX3, and the pps_act_cbcr_qp_offset_plusX4 are not signaled through a PPS syntax table.

[668] A second value (e.g., 1) of the pps_act_qp_offsets_present_flag may indicate that the pps_act_y_qp_offset, the pps_act_cb_qp_offset, and the pps_act_cr_qp_offset are signaled through a PPS syntax table.

[669] When the pps_act_qp_offsets_present_flag is not provided from a bitstream, the pps_act_qp_offsets_present_flag may be derived as the first value (e.g., 0). For example, when a flag (e.g., sps_act_enabled_flag signaled in an SPS) indicating whether ACT is applied has a first value (e.g., 0) indicating that ACT is not applied, the pps_act_qp_offsets_present_flag may be forced to have a first value (e.g., 0).

[670] When a value of a syntax element cu_act_enabled_flag is a second value (e.g., 1) indicating that ACT is applied for the current encoding basis, the syntax elements pps_act_y_qp_offset_plusX1, pps_act_cb_qp_offset_plusX2, pps_act_cr_qp_offset_plusX3, and pps_act_cbcr_qp_offset_plusX4 may be used to determine offsets applied to quantization parameter values qP for luma, Cb, Cr components, and a joint CbCr component, respectively. When the values of the pps_act_y_qp_offset_plusX1, the pps_act_cb_qp_offset_plusX2, the pps_act_cr_qp_offset_plusX3, and the pps_act_cbcr_qp_offset_plusX4 are not present in a bitstream, each value may be set to 0.

Date Reçue/Date Received 2023-12-29

[671] According to the syntax elements, values of variables PpsActQpOffsetY, PpsActQpOffsetCb, PpsActQpOffsetCr, and PpsActQpOffsetCbCr may be determined as shown in the equation below.

[672] [Equation 75]

[673] PpsActQpOffsetY = pps_act_y_qp_offset_plusX1 – X1

[674] PpsActQpOffsetCb = pps_act_cb_qp_offset_plusX2 –X2

[675] PpsActQpOffsetCr = pps_act_cr_qp_offset_plusX3 – X3

[676] PpsActQpOffsetCbCr = pps_act_cbcr_qp_offset_plusX4 – X4

[677] Herein, X1, X2, X3, and X4 may indicate predetermined constant values. These may be the same values or different values, or only some may have the same value.

[678] In an embodiment, for bitstream matching, the values of the PpsActQpOffsetY, the PpsActQpOffsetCb, the PpsActQpOffsetCr, and the PpsActQpOffsetCbCr may be limited to have values ranging from -12 to 12.

[679] According to determination of the variables, a quantization parameter qP may be determined as follows. First, when the cIdx has a value of 0, the qP and the ACT Qp offset may be derived as shown in the equation below.

[680] [Equation 76]

[681] $qP = Qp'_{Y}$

[682] ActQpOffset = PpsActQpOffsetY

[683] Alternatively, when the TuCResMode[xTbY][yTbY] has a value of 2, the qP and the ACT Qp offset may be derived as shown in the equation below.

[684] [Equation 77]

[685] $qP = Qp'_{CbCr}$

[686] ActQpOffset = PpsActQpOffsetCbCr

[687] Alternatively, when the cIdx has a value of 1, the qP and the ACT Qp offset may be derived as shown in the equation below.

[688] [Equation 78]

[689] $qP = Qp'_{Cb}$

[690] ActQpOffset = PpsActQpOffsetCb

[691] Alternatively, when the cIdx has a value of 2, the qP and the ACT Qp offset may be derived as shown in the equation below.

[692] [Equation 79]

[693] $qP = Qp'_{Cr}$

[694] ActQpOffset = PpsActQpOffsetCr

[695] In another embodiment of signaling an ACT Qp offset, a plurality of ACT QP offsets may be signaled for different joint CbCr modes identified as mode A and mode B.

[696] The joint CbCr mode A may refer to a jointCbCr mode having tu_cbf_cb having a non-zero value, such as Mode 1 and Mode 2 of Table 2 described above. The joint CbCr mode

Date Reçue/Date Received 2023-12-29

B may refer to a jointCbCrmode having tu_cbf_cb having a value of 0, such as Mode 3 of Table 2 described above. The syntax table changed accordingly is shown in FIG. 42. The syntax elements indicated in the syntax table of FIG. 42 will be described.

[697] When a value of a syntax element cu_act_enabled_flag is a second value (e.g., 1) indicating that ACT is applied for the current encoding basis, syntax elements pps_act_y_qp_offset_plusX1, pps_act_cb_qp_offset_plusX2, pps_act_cr_qp_offset_plusX3, pps_act_cbcr_qp_offset_modeA_plusX4, and pps_act_cbcr_qp_offset_modeB_plusX5 may be used to determine offsets applied to quantization parameter values qP for luma, Cb, Cr components, and a joint CbCr component, respectively. When the values of the pps_act_y_qp_offset_plusX1, the pps_act_cb_qp_offset_plusX2, the pps_act_cr_qp_offset_plusX3, the pps_act_cbcr_qp_offset_modeA_plusX4, and the pps_act_cbcr_qp_offset_modeB_plusX5 are not present in a bitstream, each value may be set to 0.

[698] According to the syntax elements, values of variables PpsActQpOffsetY, PpsActQpOffsetCb, PpsActQpOffsetCr, PpsActQpOffsetCbCrModeA, and PpsActQpOffsetCbCrModeB may be determined as shown in the equation below.

[699] [Equation 80]

[700] PpsActQpOffsetY = pps_act_y_qp_offset_plusX1 – X1

[701] PpsActQpOffsetCb = pps_act_cb_qp_offset_plusX2 –X2

[702] PpsActQpOffsetCr = pps_act_cr_qp_offset_plusX3 – X3

[703] PpsActQpOffsetCbCrModeA = pps_act_cbcr_qp_offset_modeA_plusX4 – X4

[704] PpsActQpOffsetCbCrModeB = pps_act_cbcr_qp_offset_modeB_plusX5 – X5

[705] Herein, X1, X2, X3, X4, and X5 may indicate predetermined constant values. These may be the same values or different values, or only some may have the same value. In an embodiment, for bitstream matching, the values of the PpsActQpOffsetY, the PpsActQpOffsetCb, the PpsActQpOffsetCr, the PpsActQpOffsetCbCrModeA, and the PpsActQpOffsetCbCrModeB may be limited to have values ranging from -12 to 12.

[706] According to determination of the variables, a quantization parameter qP may be determined as follows. First, when the cIdx has a value of 0, the qP and the ACT Qp offset may be derived as shown in the equation below.

[707] [Equation 81]

[708] $qP = Qp'_{Y}$

[709] ActQpOffset = PpsActQpOffsetY

[710] Alternatively, when the TuCResMode[xTbY][yTbY] has a value of 2, the qP may be derived as shown in the equation below.

[711] [Equation 82]

[712] $qP = Qp'_{CbCr}$

Date Reçue/Date Received 2023-12-29

[713] Alternatively, when the cIdx has a value of 1, the qP and the ACT Qp offset may be derived as shown in the equation below.

[714] [Equation 83]

[715] $qP = Qp'_{Cb}$

[716] ActQpOffset = PpsActQpOffsetCb

[717] Alternatively, when the cIdx has a value of 2, the qP and the ACT Qp offset may be derived as shown in the equation below.

[718] [Equation 84]

[719] $qP = Qp'_{Cr}$

[720] ActQpOffset = PpsActQpOffsetCr

[721] In addition, when the cIdx does not have a value of 0 and the TuCResMode[xTbY][yTbY] does not have a value of 0, the ACT Qp offset may be derived as shown in the equation below.

[722] [Equation 85]

[723] ActQpOffset = (tu_cbf_cb[xTbY][yTbY])? PpsActQpOffsetCbCrModeA: PpsActQpOffsetCbCrModeB

[724] In the meantime, in another embodiment, when the TuCResMode[xTbY][yTbY] has a value of 2, the ActQpOffset may be derived as shown in the equation below.

[725] [Equation 86]

[726] ActQpOffset = (tu_cbf_cb[xTbY][yTbY])? (PPsQpOffsetCbCrModeA+ slice_act_CbCr_qp_offset_ModeA): (PPsQpOffsetCbCrModeB + slice_act_CbCr_qp_offset_ModeB)

[727] In another embodiment of signaling an ACT Qp offset, only ACT QP offsets for Y, Cb, and Cr may be signaled as in the syntax table of FIG. 43. An ACT QP offset for joint CbCr may be derived from PpsActQpOffsetY, PpsActQpOffsetCb, and/or PpsActQpOffsetCr.

[728] In an embodiment, an ACT Qp offset for CbCr may be set to a value of PpsActQpOffsetCb. In another embodiment, an ACT Qp offset for CbCr may be set to the same value as PpsActQpOffsetCb in the case of the joint CbCr mode in which the tu_cbf_cb has a non-zero value, or may be set to the same value as PpsActQpOffsetCr in the case of the joint CbCr mode in which the tu_cbf_cb has a value of 0. Alternatively, it may be set vice versa.

[729] FIG. 43 is a view showing another embodiment of a syntax table in which an ACT Qp offset is signaled in a PPS. According to determination of the syntax elements of FIG. 43, a quantization parameter qP may be determined as follows. First, when the cIdx has a value of 0, the qP and the ACT Qp offset may be derived as shown in the equation below.

[730] [Equation 87]

[731] $qP = Qp'_{Y}$

[732] ActQpOffset = PpsActQpOffsetY

Date Reçue/Date Received 2023-12-29

[733] Alternatively, when the TuCResMode[xTbY][yTbY] has a value of 2, the qP and the ACT Qp offset may be derived as shown in the equation below.

[734] [Equation 88]

[735] $qP = Qp'_{CbCr}$

[736] ActQpOffset = (cIdx==1)? PpsActQpOffsetCb: PpsActQpOffsetCr

[737] In the meantime, in another embodiment, a value of an ActQpOffset may be determined as follows.

[738] [Equation 89]

[739] ActQpOffset = (tu_cbf_cb[xTbY][yTbY])? PpsActQpOffsetCb: PpsActQpOffsetCr

[740] Alternatively, when the cIdx has a value of 1, the qP and the ACT Qp offset may be derived as shown in the equation below.

[741] [Equation 90]

[742] $qP = Qp'_{Cb}$

[743] ActQpOffset = PpsActQpOffsetCb

[744] Alternatively, when the cIdx has a value of 2, the qP and the ACT Qp offset may be derived as shown in the equation below.

[745] [Equation 91]

[746] $qP = Qp'_{Cr}$

[747] ActQpOffset = PpsActQpOffsetCr

[748] <u>Embodiment 7 : signaling of ACT Qp offset at multiple levels</u>

[749] In an embodiment, an ACT QP offset may be signaled at a plurality of levels. In addition to the signaling of an ACT QP offset at one level, such as a PPS, as in Embodiment 6 above, an ACT QP offset may be signaled at a lower level (e.g., a slice header, a picture header, or headers of other types suitable for Qp control).

[750] Hereinafter, two embodiments will be described. FIGS. 44 and 45 show examples in which an ACT QP offset is signaled through a slice header and a picture header. In this way, an ACT QP offset may be signaled at multiple levels.

[751] Hereinafter, the syntax elements shown in FIGS. 44 and 45 will be described. A syntax element pps_slice_act_qp_offsets_present_flag may indicate whether syntax elements slice_act_y_qp_offset, slice_act_cb_qp_offset, slice_act_cr_qp_offset, and slice_act_cbcr_qp_offset, which will be described later, are present in a slice header.

[752] For example, a first value (e.g., 0) of the pps_slice_act_qp_offsets_present_flag may indicate that the slice_act_y_qp_offset, the slice_act_cb_qp_offset, the slice_act_cr_qp_offset, and the slice_act_cbcr_qp_offset are not present in a slice header.

[753] For example, a second value (e.g., 1) of the pps_slice_act_qp_offsets_present_flag may indicate that the slice_act_y_qp_offset, the slice_act_cb_qp_offset, the slice_act_cr_qp_offset, and the slice_act_cbcr_qp_offset are present in a slice header.

Date Reçue/Date Received 2023-12-29

[754] The syntax elements slice_act_y_qp_offset, slice_act_cb_qp_offset, slice_act_cr_qp_offset, and slice_act_cbcr_qp_offset may indicate offsets for quantization parameter values qP for the luma, Cb, Cr components, and a joint CbCr component, respectively. The values of the slice_act_y_qp_offset, the slice_act_cb_qp_offset, the slice_act_cr_qp_offset, and the slice_act_cbcr_qp_offset may be limited to have values ranging from -12 to 12. When the values of the slice_act_y_qp_offset, the slice_act_cb_qp_offset, the slice_act_cr_qp_offset, and the slice_act_cbcr_qp_offset are not present in a bitstream, each value may be set to 0. Values of PpsActQpOffsetY + slice_act_y_qp_offset, PpsActQpOffsetCb + slice_act_cb_qp_offset, PpsActQpOffsetCr + slice_act_cr_qp_offset, and PpsActQpOffsetCbCr + slice_act_cbcr_qp_offset may be limited to also have values ranging from -12 to 12.

[755] Various modified embodiments of signaling an ACT QP offset for joint CbCr at a PPS level may be applied. For example, one QP offset may be signaled for joint CbCr, a plurality of ACT Qp offsets may be signaled for joint CbCr of different modes, or without signaling an ACT Qp offset for joint CbCr, a method of deriving this by using ACTQpOffsets for Y, Cb, and Cr and/or a mode of jointCbCr may be applied when signaling through a slice header is performed.

[756] Two modified embodiments are shown in FIGS. 46 and 47. FIG. 46 shows an embodiment in which an ACT Qp offset is signaled in a slice header. FIG. 47 shows another embodiment in which an ACT Qp offset is signaled in a slice header. In FIG. 47, only ACT Qp offsets for Y, Cb, and Cr may be signaled, and an ACT QP offset at a slice level for joint CbCr may be derived from slice_act_y_qp_offset, slice_act_cb_qp_offset, and/or slice_act_cr_qp_offset. This may be determined on the basis of a mode type of jointCbCr. In an embodiment, a slice-level ACT Qp offset for CbCr may be set to the same value as the slice_act_cb_qp_offset. In another embodiment, in the case of the joint CbCr mode having the tu_cbf_cb having a non-zero value, an ACT Qp offset at a slice level for joint CbCr may be set to the same value as the slice_act_cb_qp_offset. In addition, in the case of the jointCbCr mode having the tu_cbf_cb having a value of 0, an ACT Qp offset at a slice level for joint CbCr may be set to the same value as the slice_act_cr_qp_offset.

[757] In the meantime, in another embodiment, a syntax element may be signaled in a slice header or a picture header. To realize this, encoding/decoding may be performed as follows.

[758] - A flag pps_picture_slice_act_qp_offsets_present_flag indicating whether an ACT Qp offset is present in a picture header or a slice header may be signaled in a PPS.

[759] - When ACT is applicable and the value of the pps_picture_slice_act_qp_offsets_present_flag is a second value (e.g., 1), a flag pic_act_qp_offsets_present_flag indicating whether an ACT Qp offset is present in a picture header is signaled in a picture header. Herein, a second value (e.g., 1) of the

Date Reçue/Date Received 2023-12-29

pic_act_qp_offsets_present_flag may indicate that ACT Qp offsets for all slices of a picture corresponding to the picture header are provided in the picture header.

[760] - A first value (e.g., 0) of the pic_act_qp_offsets_present_flag may indicate that ACT Qp offsets for all slices of a picture corresponding to the picture header are not provided in the picture header. For example, when ACT is applicable and the value of the pps_picture_slice_act_qp_offsets_present_flag is the second value (e.g., 1) and the value of the pic_act_qp_offsets_present_flag is the first value (e.g., 0), an ACT Qp offset for a slice may be provided in a slice header.

[761] FIG. 48 is a view showing a syntax table of a PPS in which pps_pic_slice_act_qp_offsets_present_flag is signaled. The syntax element pps_pic_slice_act_qp_offsets_present_flag may indicate whether an ACT Qp offset is provided in a picture header and/or a slice header. For example, a first value (e.g., 0) of the pps_pic_slice_act_qp_offsets_present_flag may indicate that an ACT Qp offset is not provided in a picture header and a slice header. A second value (e.g., 1) of the pps_pic_slice_act_qp_offsets_present_flag may indicate that an ACT Qp offset is provided in a picture header or a slice header. When the pps_pic_slice_act_qp_offsets_present_flag is not provided in a bitstream, the value of the pps_pic_slice_act_qp_offsets_present_flag may be determined to be the first value (e.g., 0).

[762] FIG. 49 is a view showing a syntax table of a picture header for signaling an ACT Qp offset. A syntax element pic_act_qp_offsets_present_flag may indicate whether an ACT Qp offset is provided in a picture header. A first value (e.g., 0) of the pic_act_qp_offsets_present_flag may indicate that an ACT Qp offset is not provided in a picture header, but in a slice header. A second value (e.g., 1) of the pic_act_qp_offsets_present_flag may indicate that an ACT Qp offset is provided in a picture header. When the value of the pic_act_qp_offsets_present_flag is not provided in a bitstream, the value may be determined to be 0.

[763] FIG. 50 is a view showing a syntax table of a slice header for signaling an ACT Qp offset. In the syntax table of FIG. 50, syntax elements slice_act_y_qp_offset, slice_act_cb_qp_offset, slice_act_cr_qp_offset, and slice_act_cbcr_qp_offset may indicate offsets for quantization parameter values qP for luma, Cb, and Cr components. The values of the slice_act_y_qp_offset, the slice_act_cb_qp_offset, the slice_act_cr_qp_offset, and the slice_act_cbcr_qp_offset may have values ranging from -12 to 12. In addition, values of PpsActQpOffsetY + slice_act_y_qp_offset, PpsActQpOffsetCb + slice_act_cb_qp_offset, and PpsActQpOffsetCr + slice_act_cr_qp_offset may be limited to also have a range of values from -12 to 12.

[764] In the meantime, in the case in which the values of the slice_act_y_qp_offset, the slice_act_cb_qp_offset, the slice_act_cr_qp_offset, and the slice_act_cbcr_qp_offset are not provided in a bitstream, when the value of the pps_pic_slice_act_qp_offsets_present_flag is

Date Reçue/Date Received 2023-12-29

the first value (e.g., 0), the values of the slice_act_y_qp_offset, the slice_act_cb_qp_offset, and the slice_act_cr_qp_offset may be determined to be 0. Alternatively, when the value of the pps_pic_slice_act_qp_offsets_present_flag is the second value (e.g., 1), the values of the slice_act_y_qp_offset, the slice_act_cb_qp_offset, and the slice_act_cr_qp_offset may be determined to be the same values as pps_act_y_qp_offset, pps_act_cb_qp_offset, and pps_act_cr_qp_offset, respectively.

[765] In the meantime, when an ACT Qp offset is present in both a slice header and a picture header, a final offset value used to derive a qP value may be determined to be the value that is obtained by adding an offset value signaled in a PPS and an offset value signaled in the slice header or the picture header.

[766] More specifically, in an embodiment, a quantization parameter qP may be determined as follows. First, when the cIdx has a value of 0, the qP and the ACT Qp offset may be derived as shown in the equation below.

[767] [Equation 92]

[768] $qP = Qp'_{Y}$

[769] ActQpOffset = PPsQpOffsetY + slice_act_y_qp_offset

[770] Alternatively, when the TuCResMode[xTbY][yTbY] has a value of 2, the qP and the ACT Qp offset may be derived as shown in the equation below.

[771] [Equation 93]

[772] $qP = Qp'_{CbCr}$

[773] ActQpOffset = PPsQpOffsetCbCr + slice_act_CbCr_qp_offset

[774] Alternatively, when the cIdx has a value of 1, the qP and the ACT Qp offset may be derived as shown in the equation below.

[775] [Equation 94]

[776] $qP = Qp'_{Cb}$

[777] ActQpOffset = PpsActQpOffsetCb + slice_act_Cb_qp_offset

[778] Alternatively, when the cIdx has a value of 2, the qP and the ACT Qp offset may be derived as shown in the equation below.

[779] [Equation 95]

[780] $qP = Qp'_{Cr}$

[781] ActQpOffset = PpsActQpOffsetCr + slice_act_Cr_qp_offset

[782] In another embodiment, when multiple ACT Qp offsets for joint CbCr are signaled, ActQpOffset for joint CbCr may be determined as follows.

[783] First, when the cIdx has a value of 0, the qP and the ACT Qp offset may be derived as shown in the equation below.

[784] [Equation 96]

[785] $qP = Qp'_{Y}$

[786] ActQpOffset = PPsQpOffsetY + slice_act_y_qp_offset

Date Reçue/Date Received 2023-12-29

[787] Alternatively, when the TuCResMode[xTbY][yTbY] has a value of 2, the qP may be derived as shown in the equation below.

[788] [Equation 97]

[789] $qP = Qp'_{CbCr}$

[790] Alternatively, when the cIdx has a value of 1, the qP and the ACT Qp offset may be derived as shown in the equation below.

[791] [Equation 98]

[792] $qP = Qp'_{Cb}$

[793] ActQpOffset = PpsActQpOffsetCb + slice_act_Cb_qp_offset

[794] Alternatively, when the cIdx has a value of 2, the qP and the ACT Qp offset may be derived as shown in the equation below.

[795] [Equation 99]

[796] $qP = Qp'_{Cr}$

[797] ActQpOffset = PpsActQpOffsetCr + slice_act_Cr_qp_offset

[798] In addition, when the cIdx does not have a value of 0 and the TuCResMode[xTbY][yTbY] does not have a value of 0, the ACT Qp offset may be derived as shown in the equation below.

[799] [Equation 100]

[800] ActQpOffset = (tu_cbf_cb[xTbY][yTbY])? (PPsQpOffsetCbCrModeA+ slice_act_CbCr_qp_offset_ModeA): (PPsQpOffsetCbCrModeB + slice_act_CbCr_qp_offset_ModeB)

[801] In still another embodiment, when an ACT Qp offset for joint CbCr is not provided, qP and ActQpOffset for Y, Cb, and/or Cr components are determined and the ActQpOffset for joint CbCr may be determined using ACT Qp offsets of the Y, Cb, and/or Cr components as follows. For example, in the above-described embodiment, when the TuCResMode[xTbY][yTbY] related to Equation 97 has a value of 2, the calculation step of the qP may be changed and performed as follows.

[802] "Alternatively, when the TuCResMode[xTbY][yTbY] has a value of 2, the qP and the ACT Qp offset may be derived as shown in the equation below.

[803] [Equation 101]

[804] $qP = Qp'_{CbCr}$

[805] ActQpOffset = (cIdx==1])? (PPsQpOffsetCb + slice_act_Cb_qp_offset): (PPsQpOffsetCr + slice_act_Cr_qp_offset)"

[806] In the meantime, in another embodiment, a value of ActQpOffset may be determined as in the equation below.

[807] [Equation 102]

[808] ActQpOffset = (tu_cbf_cb[xTbY][yTbY])? (PPsQpOffsetCb + slice_act_Cb_qp_offset): (PPsQpOffsetCr + slice_act_Cr_qp_offset)

Date Reçue/Date Received 2023-12-29

[809] Embodiment 8 : method of signaling multiple ACT Qp offset sets

[810] In the present embodiment, a method of using a list of ACT Qp offsets will be described. To this end, the following processing may be performed.

[811] a) Multiple sets of ACT Qp offsets may be signaled in the form of a list within a parameter set (e.g., SPS or PPS). Each set in the list may include ACT Qp offsets for Y, Cb, Cr, and joint CbCr components. For simplicity, the list of ACT Qp offsets may be signaled in the parameter set the same as the parameter set for signaling a list of chroma Qp offsets.

[812] b) The number of the sets of the ACT Qp offsets in the list may be the same as the number of chroma Qp offset sets signaled in a PPS.

[813] c) As an ACT Qp offset used to derive the qP for each encoding basis, an ACT Qp offset belonging to the list having an index (e.g., cu_chroma_qp_offset_idx) for a chroma Qp offset for the encoding basis may be used.

[814] d) As an alternative embodiment of b) and c), the following may be performed.

[815] - The number of the sets of the ACT Qp offsets in the list may be signaled. The number of the sets of the ACT Qp offsets in the list may be different from the number of chroma Qp offset sets.

[816] - When ACT is applicable, an index indicating the index of the ACT Qp offset used for the encoding basis may be signaled.

[817] Without departing from the above concept, the syntax for signaling a list of ACT Qp offsets may be used as shown in FIG. 51. For example, when cu_act_enabled_flag has a value of 1, pps_act_y_qp_offset, pps_act_cb_qp_offset, pps_act_cr_qp_offset, and pps_act_cbcr_qp_offset may be used to determine offsets to be applied to quantization parameter values qP for luma, Cb, Cr components, and joint CbCr, respectively.

[818] When the values of the pps_act_y_qp_offset, the pps_act_cb_qp_offset, the pps_act_cr_qp_offset, and the pps_act_cbcr_qp_offset are not present, each value may be derived into 0.

[819] When the value of the cu_act_enabled_flag is a second value (e.g., 1) and a value of cu_chroma_qp_offset_flag is a second value (e.g., 1), act_y_qp_offset_list[i], act_cb_qp_offset_list[i], act_cr_qp_offset_list[i], and act_cbcr_qp_offset_list[i] may be used to determine offsets applied to quantization parameters value qP for luma, Cb, Cr components, and a joint CbCr component, respectively. When the values of the act_y_qp_offset_list[i], the act_cb_qp_offset_list[i], the act_cr_qp_offset_list[i], and the act_cbcr_qp_offset_list[i] are not present, each value may be derived into 0.

[820] In the present embodiment, a quantization parameter qP may be determined as follows. First, when the cIdx has a value of 0, the qP and the ACT Qp offset may be derived as shown in the equation below.

[821] [Equation 103]

[822] $qP = Qp'_Y$

Date Reçue/Date Received 2023-12-29

[823] ActQpOffset = pps_act_y_qp_offset + (cu_chroma_qp_offset_flag) ? act_y_qp_offset_list[cu_chroma_qp_offset_idx] : 0 + slice_act_y_qp_offset

Alternatively, when the TuCResMode[xTbY][yTbY] has a value of 2, the qP and the ACT Qp offset may be derived as shown in the equation below.

[824] [Equation 104]

[825] $qP = Qp'_{CbCr}$

[826] ActQpOffset = pps_act_cbcr_qp_offset + (cu_chroma_qp_offset_flag) ? act_cbcr_qp_offset_list[cu_chroma_qp_offset_idx] : 0 + slice_act_cbcr_qp_offset

[827] Alternatively, when the cIdx has a value of 1, the qP and the ACT Qp offset may be derived as shown in the equation below.

[828] [Equation 105]

[829] $qP = Qp'_{Cb}$

[830] ActQpOffset = pps_act_cb_qp_offset + (cu_chroma_qp_offset_flag) ? act_cb_qp_offset_list[cu_chroma_qp_offset_idx] : 0 + slice_act_cb_qp_offset

Alternatively, when the cIdx has a value of 2, the qP and the ACT Qp offset may be derived as shown in the equation below.

[831] [Equation 106]

[832] $qP = Qp'_{Cr}$

[833] ActQpOffset = pps_act_cr_qp_offset + (cu_chroma_qp_offset_flag) ? act_cr_qp_offset_list[cu_chroma_qp_offset_idx] : 0 + slice_act_cr_qp_offset

Embodiment 9 : ACT color space transform method to be applied to both lossless encoding and lossy encoding

[834] Transform between color spaces based on the matrices for forward transform and backward transform described above may be organized as follows.

[835] [Table 3]

Forward Conversion: GBR to YCgCo	Backward Conversion: YCgCo to GBR
Y = ((G<<1) + R + B + 2) >>.2 Cg = ((G<<1) - R – B + 2) >>2 Co = ((R-B) << 1) + 2) >> 2	t = Y – Cg G = Y + Cg B = t – Co = Y – Cg – Co R = t + Co = Y – Cg + Co

[836] The transform is unable to achieve reconstruction into the original state because loss of some values occurs in the processing of Co and Cg. For example, when a sample value of the RGB color space is converted into that of the YCgCocolor space and the resulting value is inversely converted back to that of the RGB color space, the value of the original sample is not completely reconstructed. Therefore, the transform according to Table 3 cannot be used for lossless encoding. It is necessary to improve a color space transform algorithm so that loss of a sample value does not occur after color space transform even when lossless encoding is

Date Reçue/Date Received 2023-12-29

applied. Embodiments 9 and 10 describe a color space transform algorithm capable of being applied to lossless encoding as well as lossy encoding.

[837] In the embodiment described below, described is a method of performing ACT by using color space transform reversible into an original state, which is applicable to lossy encoding as well as lossless encoding. This reversible color space transform may be applied to the above-described encoding and decoding methods. An ACT Qp offset may also be adjusted for the color space transform below. Color space transform according to an embodiment may be performed as shown in the equation below. For example, forward transform from a GBR color space to a YCgCo color space may be performed according to the equation below.

[838] [Equation 107]

[839] $Co = R-B;$

[840] $t = B + (Co>>1);$

[841] $Cg = G - t;$

[842] $Y = t + (Cg>>1);$

[843] In addition, backward transform from a YCgCo color space to a GBR color space may be performed according to the equation below.

[844] [Equation 108]

[845] $t = Y - (Cg>>1)$

[846] $G = Cg + t$

[847] $B = t - (Co>>1)$

[848] $R = Co + B$

[849] The transform between the YCgCo color space and the RGB color space according to the equations above is reversible to an original state. That is, the color space transform according to the equations supports perfect reconstruction. For example, a sample value is maintained even though backward transform is performed after forward transform. Accordingly, the color space transform according to the equations may be referred to as reversible YCgCo-R color transform. Herein, R may stand for reversible, which means that achieves reconstruction to an original state. YCgCo-R transform may be provided by increasing the bit depths of Cg and Co by 1 compared to the existing transform. If this condition is met, other types of reversible transform may be used like the above-described transform.

[850] Because the transform as shown in the equations above has a different norm value from the above-described transform, ACT Qp offsets for Y, Cg, and Co may be adjusted to compensate for a dynamic range change due to color space transform.

[851] It has been described that when the above-described transform is applied, a QCT Qp offset according to an embodiment may have a value of (-5, -5, -5) for Y, Cg, and Co. However, when the reversible transform according to the present embodiment is applied, as a QCT Qp offset according to an embodiment, another value not (-5, -5, -5) may be specified. For example,

Date Reçue/Date Received 2023-12-29

as a QCT Qp offset according to an embodiment, a value of (-5, 1, 3) for Y, Cg, and Co may be used.

[852] In another embodiment, an ACT QP offset may be signaled through a bitstream as in Embodiment 6 or 7 described above.

[853] For example, when the YCgCo-R transform described above was used together with an ACT QP offset of (-5, 1, 3), it was observed that encoding loss was not present in a lossy encoding environment (e.g., QP 22, 27, 32, 37), as shown in the figure below. In addition, it was observed that when ACT is applied, encoding performance of 5 % was further obtained in realizing lossless encoding.

[854] [Table 4]

Sequence	Y	U	V
RGB, TGM 1080p	0.0%	0.2%	0.1%
RGB, TGM 720p	0.2%	-0.1%	0.1%
RGB, Animation	-0.1%	-0.1%	0.0%
RGB, Mixed content	-0.1%	0.0%	-0.1%
RGB, Camera-Captured content	-0.3%	0.2%	-0.3%
Overall All (RGB)	0.0%	0.0%	0.0%

[855] The VVC specification for including an integrated ACT matrix may be described as the table below.

[856] [Table 5]

Residual modification process for blocks using colour space conversion

Inputs to this process are:

– a variable nTbW specifying the block width,

– a variable nTbH specifying the block height,

– an (nTbW)x(nTbH) array of luma residual samples r_Y with elements $r_Y[x][y]$,

– an (nTbW)x(nTbH) array of chroma residual samples r_{Cb} with elements $r_{Cb}[x][y]$,

– an (nTbW)x(nTbH) array of chroma residual samples r_{Cr} with elements $r_{Cr}[x][y]$.

Outputs of this process are:

– a modified (nTbW)x(nTbH) array r_Y of luma residual samples,

– a modified (nTbW)x(nTbH) array r_{Cb} of chroma residual samples,

– a modified (nTbW)x(nTbH) array r_{Cr} of chroma residual samples.

The (nTbW)x(nTbH) arrays of residual samples r_Y, r_{Cb} and r_{Cr} are modified as follows:

$tmp = r_Y[x][y] – (r_{Cb}[x][y] >>1)$

$r_Y[x][y] = tmp + r_{Cb}[x][y]$

$r_{Cb}[x][y] = tmp – (r_{Cr}[x][y] >>1)$

Date Reçue/Date Received 2023-12-29

$r_{Cr}[x][y] = r_{Cb}[x][y] + r_{Cr}[x][y]$

[857] For example, (nTbW)x(nTbH)-sized residual sample arrays r_Y, r_{Cb}, and r_{Cr} may be updated as follows. [Equation 109]

[858] $tmp = r_Y[x][y] – (r_{Cb}[x][y] >>1)$

[859] $r_Y[x][y] = tmp + r_{Cb}[x][y]$

[860] $r_{Cb}[x][y] = tmp – (r_{Cr}[x][y] >>1)$

[861] $r_{Cr}[x][y] = r_{Cb}[x][y] + r_{Cr}[x][y]$

[862] Embodiment 10 : ACT performance method for performing multiple color transforms based on explicit signaling

[863] In the present embodiment, at least one color transform may be performed by ACT. Which color transform is to be performed may be determined by a flag(s) signaled in a bitstream. The flag(s) may be signaled at multiple levels, such as an SPS, a PPS, a picture header, and a slice, or identifiable granularities.

[864] In an embodiment, a predetermined flag may be signaled to indicate which ACT is applied. For example, when the flag has a value of 1, ACT based on reversible color transform may be applied. When the flag has a value of 0, ACT based on irreversible color transform may be applied.

[865] In another embodiment, a predetermined flag for ACT may be signaled to indicate which color transform is used. An example of syntax signaled in an SPS is shown in FIG. 52. The syntax element of FIG. 52 will be described. A syntax element sps_act_reversible_conversion may indicate whether a conversion equation unreversible to an original state is used. A first value (e.g., 0) of the sps_act_reversible_conversion may indicate that ACT uses a conversion equation unreversible to an original state. A second value (e.g., 1) of the sps_act_reversible_conversion may indicate that ACT uses a transform equation reversible to an original state.

[866] Accordingly, a variable lossyCoding indicating whether lossy encoding is performed may be set as in the equation below.

[867] [Equation 110]

[868] lossyCoding = (!sps_act_reversible_conversion)

[869] By using the lossyCoding flag, the pseudocode for the decoding device to perform backward transform from YCgCo to GBR in the decoding process may be expressed as follows.

[870] [Equation 111]

[871] If (sps_act_reversible_conversion == 1)

[872] {

[873] // YCgCo-R reversible conversion

[874] t = Y – (Cg>>1)

[875] G = Cg + t

[876] B = t – (Co>>1)

Date Reçue/Date Received 2023-12-29

[877] R = Co + B

[878] }

[879] else {

[880] t = Y – Cg

[881] G = Y + Cg

[882] B = t – Co

[883] R = t + Co

[884] }

[885] Accordingly, the VVC specification shown in Table 5 of Embodiment 9 may be modified as shown in the table below.

[886] [Table 6]

Residual modification process for blocks using colour space conversion

Inputs to this process are:

– a variable nTbW specifying the block width,

– a variable nTbH specifying the block height,

– an (nTbW)x(nTbH) array of luma residual samples r_Y with elements $r_Y[x][y]$,

– an (nTbW)x(nTbH) array of chroma residual samples r_{Cb} with elements $r_{Cb}[x][y]$,

– an (nTbW)x(nTbH) array of chroma residual samples r_{Cr} with elements $r_{Cr}[x][y]$.

Outputs of this process are:

– a modified (nTbW)x(nTbH) array r_Y of luma residual samples,

– a modified (nTbW)x(nTbH) array r_{Cb} of chroma residual samples,

– a modified (nTbW)x(nTbH) array r_{Cr} of chroma residual samples.

The (nTbW)x(nTbH) arrays of residual samples r_Y, r_{Cb} and r_{Cr} are modified as follows:

- If **sps_act_ reversible _conversion** equal to 1, the (nTbW)x(nTbH) arrays of residual samples r_Y, r_{Cb} and r_{Cr} are modified as follows

$tmp = r_Y[x][y] – (r_{Cb}[x][y] >>1)$)

$r_Y[x][y] = tmp + r_{Cb}[x][y]$)

$r_{Cb}[x][y] = tmp – (r_{Cr}[x][y] >>1)$)

$r_{Cr}[x][y] = r_{Cb}[x][y] + r_{Cr}[x][y]$

- Otherwise, the (nTbW)x(nTbH) arrays of residual samples r_Y, r_{Cb} and r_{Cr} are modified as follows

$tmp = r_Y[x][y] - r_{Cb}[x][y]$)

$r_Y[x][y] = r_Y[x][y] + r_{Cb}[x][y]$)

Date Reçue/Date Received 2023-12-29

$r_{Cb}[x][y] = tmp - r_{Cr}[x][y]$)

$r_{Cr}[x][y] = tmp + r_{Cr}[x][y]$

[887] According to the table above, a residual update process using color space transform may use the following parameters as inputs to the present process. - a variable nTbW indicating a block width

[888] - a variable nTbH indicating a block height

[889] - a (nTbW)x(nTbH)-sized array r_Y for a luma residual sample, composed of elements $r_Y[x][y]$,

[890] - a (nTbW)x(nTbH)-sized array $r_{Cb}$for a chroma residual sample, composed of elements $r_{Cb}[x][y]$,

[891] - a (nTbW)x(nTbH)-sized array r_{Cr} for a chroma residual sample, composed of elements $r_{Cr}[x][y]$,

[892] The output for the present process is as follows.

[893] - a (nTbW)x(nTbH)-sized updated array r_Y for a luma residual sample,

[894] - a (nTbW)x(nTbH)-sized updated array r_{Cb} for a chroma residual sample,

[895] - a (nTbW)x(nTbH)-sized updated array r_{Cr} for a chroma residual sample,

[896] By performing the present process, (nTbW)x(nTbH)-sized residual sample arrays r_Y, r_{Cb}, and r_{Cr} may be updated as follows.

[897] First, when the value of the sps_act_reversible_conversion is the second value (e.g., 1), (nTbW) x (nTbH)-sized residual sample arrays r_Y, r_{Cb}, and r_{Cr} may be updated as shown in the equation below.

[898] [Equation 112]

[899] $tmp = r_Y[x][y] – (r_{Cb}[x][y] >>1)$)

[900] $r_Y[x][y] = tmp + r_{Cb}[x][y]$)

[901] $r_{Cb}[x][y] = tmp – (r_{Cr}[x][y] >>1)$)

[902] $r_{Cr}[x][y] = r_{Cb}[x][y] + r_{Cr}[x][y]$

[903] Otherwise (e.g., when the value of the sps_act_reversible_conversion is the first value (e.g., 0)), (nTbW) x (nTbH)-sized residual sample arrays r_Y, r_{Cb}, and r_{Cr} may be updated as shown in the equation below.

[904] [Equation 113]

[905] $tmp = r_Y[x][y] - r_{Cb}[x][y]$

[906] $r_Y[x][y] = r_Y[x][y] + r_{Cb}[x][y]$

[907] $r_{Cb}[x][y] = tmp - r_{Cr}[x][y]$

[908] $r_{Cr}[x][y] = tmp + r_{Cr}[x][y]$

[909] YCgCo backward transform and YCgCo-R backward transform have some similarities. In transform reversible to an original state, when Cg and Co are replaced with Cg'=Cg<<1 and

Date Reçue/Date Received 2023-12-29

Co'=Co<<1, this may operate as lossy backward transform. The equation below shows an embodiment thereof.

[910] [Equation 114]

[911] $t = Y - (Cg'>>1) = Y - Cg$

[912] $G = Cg' + t = Y + Cg$

[913] $B = t - (Co'>>1) = t - Co = Y - Cg - Co$

[914] $R = Co' + B = t + Co = Y - Cg + Co$

[915] Accordingly, in an alternative embodiment, instead of maintaining two color transforms, only transform reversible to an original state may be used. In the lossy encoding case, Cg and Co components may be scaled 1/2 times in the operation of the encoding device, and may be scaled twice in the operation of the decoding device. This enables one integrated transform to be used even when lossy and lossless cases are supported. In addition, there is an additional advantage that bit depth does not change even when lossy encoding is in progress.

[916] [Table 7]

Backward Conversion : YCgCo to GBR

If (lossyCoding)
{
Cg = Cg << 1;
Co = Co << 1;
}
t = Y – (Cg>>1)
G = Cg + t
B = t – (Co>>1)
R = Co + B

[917] In an embodiment, a flag (e.g., actShiftFlag) indicating which ACT transform is used may be used according to the syntax of FIG. 53. In the syntax table of FIG. 53, a syntax element sps_act_shift_flag may indicate whether performing color component shifting is applied while ACT is applied. For example, a first value (e.g., 0) of the sps_act_shift_flag may indicate that performing color component shifting is not applied while ACT is applied. A second value (e.g., 1) of the sps_act_shift_flag may indicate that performing color component shifting is applied while ACT is applied. The variable actShiftFlag may be set to the value of the sps_act_shift_flag. The pseudocode for realizing backward transform from YCgCo to GBR in the decoding device may be written using the actShiftFlag, as follows.

[918] [Table 8]

Backward Conversion: YCgCo to GBR
If (actShiftFlag) {

Date Reçue/Date Received 2023-12-29

```
    Cg  = Cg << 1;
    Co =  Co << 1;
}

t = Y – (Cg>>1)
G =  Cg + t
B = t – (Co>>1)
R = Co + B
```

[919] Encoding and decoding methods

[920] Hereinafter, an image encoding method performed by an image encoding device and an image decoding method performed by an image decoding device according to an embodiment will be described.

[921] First, the operation of the decoding device will be described. The image decoding device according to an embodiment may include a memory and a processor. The decoding device may perform decoding according to the operation of the processor. FIG. 54 shows a decoding method of a decoding device according to an embodiment. As shown in FIG. 54, the decoding device according to the embodiment may determine a residual sample of a current block in step S5410. Next, the decoding device may reset a value of the residual sample on the basis of whether transform of a color space is applied, in step S5420. Herein, transform of a color space may mean the color space transform described above. As described above, the decoding device may determine that transform of a color space is applied when a value of a flag (e.g., cu_act_enabled_flag) indicating whether transform of a color space is applied indicates that transform of a color space is applied (e.g., cu_act_enabled_flag == 1).

[922] Referring to FIG. 55, the determining of the residual sample of the current block by the decoding device according to an embodiment will be described in more detail. The decoding device according to the embodiment may determine a prediction mode of the current block on the basis of prediction information obtained from a bitstream, in step S5510. For example, the decoding device may obtain, from a bitstream, a prediction mode flag (e.g., pred_mode_flag) indicating whether a prediction mode of a current encoding basis is an inter mode or an intra mode. When its value is a first value (e.g., pred_mode_flag == 0), the prediction mode of the current block is determined to be the inter mode. When the value is a second value (e.g., pred_mode_flag == 1), the prediction mode of the current block is determined to be the intra mode.

[923] Next, the decoding device may obtain, on the basis of the prediction mode of the current block, a flag indicating whether color space transform is applied to the residual sample of the current block, in step S5520. For example, the decoding device may obtain cu_act_enabled_flag from a bitstream on the basis of whether the prediction mode of the current block is an intra prediction mode. For example, the decoding device may obtain the

Date Reçue/Date Received 2023-12-29

cu_act_enabled_flag from a bitstream when the prediction mode of the current block is the intra prediction mode. In an embodiment, when the prediction mode of the current block is not the intra prediction mode, the decoding device may obtain the cu_act_enabled_flag from a bitstream only for the case in which information on transform is obtained for the current encoding block. For example, only for the case in which a value of a flag (e.g., cu_coded_flag) indicating whether a syntax element for a transform block is obtained from a bitstream on the basis of transform tree syntax (e.g., transform_tree()) or transform unit syntax (e.g., transform_unit()) is a predetermined value (e.g., 1), the decoding device may obtain the cu_act_enabled_flag from a bitstream even when the prediction mode of the current block is not the intra prediction mode. For example, when a syntax element for a transform unit may be obtained from a bitstream and the prediction mode of the current encoding basis is an inter prediction mode or an IBC mode, the decoding device may obtain the cu_act_enabled_flag from a bitstream.

[924] Next, as described in the previous embodiment, the decoding device may determine a quantization parameter of the current block on the basis of the value of the cu_act_enabled_flag in step S5530. Next, the decoding device may determine a transform coefficient of the current block on the basis of the quantization parameter in step S5540. Next, the decoding device may determine the residual sample on the basis of the transform coefficient in step S5550.

[925] Herein, the decoding device may determine the quantization parameter by applying clipping to the quantization parameter such that a value of the quantization parameter has a value in a predetermined range. Herein, a lower limit value of the predetermined range may be 0. In addition, an upper limit value of the predetermined range may be determined on the basis of a syntax element indicating a bit depth of a sample. For example, the upper limit value of the predetermined range may be determined to be 63 + QpBdOffset. The QpBdOffset denotes a luma and chroma quantization parameter range offset, and may be preset to a predetermined constant or obtained from a bitstream. For example, the QpBdOffset may be calculated by multiplying a predetermined constant (e.g., 6) by a value of a syntax element (e.g., sps_bitdepth) indicating a bit depth of a luma or chroma sample.

[926] In addition, the decoding device may determine the quantization parameter on the basis of a color component of the current block, may determine a quantization parameter offset on the basis of the color component of the current block, and may then reset the quantization parameter by using the quantization parameter offset, thereby determining the quantization parameter. For example, the decoding device may add the quantization parameter offset (e.g., ActQpOffset) to the quantization parameter (e.g., qP), thereby resetting the quantization parameter.

[927] The quantization parameter offset may be determined as follows. When color space transform is applied to the residual sample of the current block and the color component of the current block is a luma component, a value of the quantization parameter offset may be

Date Reçue/Date Received 2023-12-29

determined to be -5. Alternatively, when color space transform is applied to the residual sample of the current block and the color component of the current block is a chroma Cb component, a value of the quantization parameter offset may be determined to be 1. Alternatively, when color space transform is applied to the residual sample of the current block and the color component of the current block is a chroma Cr component, a value of the quantization parameter offset may be determined to be 3.

[928] Hereinafter, the step S5420 of resetting the value of the residual sample on the basis of whether transform of a color space is applied will be described in detail. The resetting of the value of the residual sample may be performed on the basis of a half value of a chroma residual sample value. For example, this may be performed as in Equation 109 of Embodiment 9 described above. For example, the half value of the chroma residual sample value may be derived by applying right-shift operation to the chroma residual sample value. For example, the resetting of the value of the residual sample may be performed on the basis of the half value of the chroma residual sample value and a luma component residual sample value. For example, the resetting of the value of the residual sample may be performed on the basis of the sum of the half value of the chroma residual sample value and the luma component residual sample value. In an embodiment, the resetting of the value of the residual sample may be performed on the basis of the sum of the luma component residual sample value and a half value of a chroma Cb component residual sample value.

[929] In an embodiment, the luma component residual sample value of the residual sample may be reset on the basis of adding the half value of the chroma Cb component residual sample to the luma component residual sample value. Herein, the half value of the chroma Cb component residual sample added to the luma component residual sample value may be determined on the basis of subtracting, from the chroma Cb component residual sample value, a value that is obtained by applying right-shift operation to the chroma Cb component residual sample value.

[930] In an embodiment, the chroma Cb component residual sample value of the residual sample may be reset on the basis of subtracting, from the luma component residual sample value, the half value of the chroma Cb component residual sample and a half value of a chroma Cr component residual sample. Herein, the half value of the chroma Cb component residual sample may be determined on the basis of a value that is obtained by applying right-shift operation to the chroma Cb component residual sample value. The half value of the chroma Cr component residual sample may be determined on the basis of a value that is obtained by applying right-shift operation to the chroma Cr component residual sample value.

[931] In an embodiment, the chroma Cr component residual sample value of the residual sample may be reset on the basis of subtracting the half value of the chroma Cb component residual sample from the luma component residual sample value and adding the half value of the chroma Cr component residual sample thereto. Herein, the half value of the chroma Cb

Date Reçue/Date Received 2023-12-29

component residual sample may be determined on the basis of a value that is obtained by applying right-shift operation to the chroma Cb component residual sample value. The half value of the chroma Cr component residual sample may be determined on the basis of subtracting, from the chroma Cr component residual sample value, a value that is obtained by applying right-shift operation to the chroma Cr component residual sample value.

[932] Hereinafter, the operation of the encoding device will be described. The image encoding device according to an embodiment may include a memory and a processor. The encoding device may perform encoding according to the operation of the processor in a manner that corresponds to the decoding by the decoding device. For example, as shown in FIG. 56, the encoding device may determine a residual sample of a current block in step S5610. Next, the encoding device may reset a value of the residual sample on the basis of whether transform of a color space is applied, in step S5620. Herein, the resetting of the value of the residual sample may be performed on the basis of a half value of a chroma residual sample value so as to correspond to the decoding by the decoding device described above.

[933] Referring to FIG. 57, the operation of the encoding device will be described in more detail. The encoding device may determine a prediction mode of the current block in step S5710. For example, the encoding device may determine the prediction mode of the current block to be an intra, inter, or IBC mode, on the basis of an encoding rate. Next, the encoding device may generate a prediction block on the basis of the determined prediction mode, and may generate a residual block on the basis of a difference between an original image block and the prediction block, thereby determining the residual sample in step S5720. Next, the encoding device may reset a value of the residual sample on the basis of whether transform of a color space is applied, in step S5730. For example, the encoding device may determine whether transform of a color space is applied, on the basis of an encoding rate. When color space transform is applied, color space transform of a sample may be performed using the transform equation (e.g., Equation 107 or 108) described in Embodiments 9 and 10. For example, in order to perform inverse transform by the decoding device described above, the encoding device may perform transform of a color space according to the equation corresponding to the inverse of Equation 109.

[934] Next, the encoding device may determine a quantization parameter of the current block depending on whether color space transform is applied, in step S5740. Corresponding to the operation of the decoding device described above, the encoding device may determine the quantization parameter on the basis of a color component of the current block, may determine a quantization parameter offset on the basis of the color component of the current block, and may then reset the quantization parameter by using the quantization parameter offset, thereby determining the quantization parameter. For example, the encoding device may add the quantization parameter offset to the quantization parameter, thereby resetting the quantization parameter.

Date Reçue/Date Received 2023-12-29

[935] The quantization parameter offset may be determined as follows. When color space transform is applied to the residual sample of the current block and the color component of the current block is a luma component, a value of the quantization parameter offset may be determined to be -5. Alternatively, when color space transform is applied to the residual sample of the current block and the color component of the current block is a chroma Cb component, a value of the quantization parameter offset may be determined to be 1. Alternatively, when color space transform is applied to the residual sample of the current block and the color component of the current block is a chroma Cr component, a value of the quantization parameter offset may be determined to be 3.

[936] In addition, the encoding device may determine the quantization parameter by applying clipping to the quantization parameter such that a value of the quantization parameter has a value in a predetermined range. Herein, a lower limit value of the predetermined range may be 0. In addition, an upper limit value of the predetermined range may be set to a predetermined upper limit value. The upper limit value may be determined on the basis of a bit depth of a sample. For example, the upper limit value of the predetermined range may be determined to be 63 + QpBdOffset. Herein, the QpBdOffset denotes a luma and chroma quantization parameter range offset, and may be preset to a predetermined constant or obtained from a bitstream. For example, the QpBdOffset may be calculated by multiplying a predetermined constant by a value of a syntax element indicating a bit depth of a luma or chroma sample.

[937] Next, the encoding device may determine a transform coefficient of the current block on the basis of the quantization parameter in step S5750. Next, the encoding device may encode, on the basis of the prediction mode of the current block, information (e.g., cu_act_enabled_flag) indicating whether color space transform is applied to the residual sample of the current block, in step S5760.

[938] <u>Application embodiment</u>

[939] While the exemplary methods of the present disclosure described above are represented as a series of operations for clarity of description, it is not intended to limit the order in which the steps are performed, and the steps may be performed simultaneously or in different order as necessary. In order to implement the method according to the present disclosure, the described steps may further include other steps, may include remaining steps except for some of the steps, or may include other additional steps except for some steps.

[940] In the present disclosure, the image encoding apparatus or the image decoding apparatus that performs a predetermined operation (step) may perform an operation (step) of confirming an execution condition or situation of the corresponding operation (step). For example, if it is described that predetermined operation is performed when a predetermined condition is satisfied, the image encoding apparatus or the image decoding apparatus may

Date Reçue/Date Received 2023-12-29

perform the predetermined operation after determining whether the predetermined condition is satisfied.

[941] The various embodiments of the present disclosure are not a list of all possible combinations and are intended to describe representative aspects of the present disclosure, and the matters described in the various embodiments may be applied independently or in combination of two or more.

[942] Various embodiments of the present disclosure may be implemented in hardware, firmware, software, or a combination thereof. In the case of implementing the present disclosure by hardware, the present disclosure can be implemented with application specific integrated circuits (ASICs), Digital signal processors (DSPs), digital signal processing devices (DSPDs), programmable logic devices (PLDs), field programmable gate arrays (FPGAs), general processors, controllers, microcontrollers, microprocessors, etc.

[943] In addition, the image decoding apparatus and the image encoding apparatus, to which the embodiments of the present disclosure are applied, may be included in a multimedia broadcasting transmission and reception device, a mobile communication terminal, a home cinema video device, a digital cinema video device, a surveillance camera, a video chat device, a real time communication device such as video communication, a mobile streaming device, a storage medium, a camcorder, a video on demand (VoD) service providing device, an OTT video (over the top video) device, an Internet streaming service providing device, a three-dimensional (3D) video device, a video telephony video device, a medical video device, and the like, and may be used to process video signals or data signals. For example, the OTT video devices may include a game console, a Blu-ray player, an Internet access TV, a home theater system, a smartphone, a tablet PC, a digital video recorder (DVR), or the like.

[944] FIG. 58 is a view showing a content streaming system, to which an embodiment of the present disclosure is applicable.

[945] As shown in FIG. 58, the content streaming system, to which the embodiment of the present disclosure is applied, may largely include an encoding server, a streaming server, a web server, a media storage, a user device, and a multimedia input device.

[946] The encoding server compresses content input from multimedia input devices such as a smartphone, a camera, a camcorder, etc. into digital data to generate a bitstream and transmits the bitstream to the streaming server. As another example, when the multimedia input devices such as smartphones, cameras, camcorders, etc. directly generate a bitstream, the encoding server may be omitted.

[947] The bitstream may be generated by an image encoding method or an image encoding apparatus, to which the embodiment of the present disclosure is applied, and the streaming server may temporarily store the bitstream in the process of transmitting or receiving the bitstream.

Date Reçue/Date Received 2023-12-29

[948] The streaming server transmits the multimedia data to the user device based on a user's request through the web server, and the web server serves as a medium for informing the user of a service. When the user requests a desired service from the web server, the web server may deliver it to a streaming server, and the streaming server may transmit multimedia data to the user. In this case, the content streaming system may include a separate control server. In this case, the control server serves to control a command/response between devices in the content streaming system.

[949] The streaming server may receive content from a media storage and/or an encoding server. For example, when the content are received from the encoding server, the content may be received in real time. In this case, in order to provide a smooth streaming service, the streaming server may store the bitstream for a predetermined time.

[950] Examples of the user device may include a mobile phone, a smartphone, a laptop computer, a digital broadcasting terminal, a personal digital assistant (PDA), a portable multimedia player (PMP), navigation, a slate PC, tablet PCs, ultrabooks, wearable devices (e.g., smartwatches, smart glasses, head mounted displays), digital TVs, desktops computer, digital signage, and the like.

[951] Each server in the content streaming system may be operated as a distributed server, in which case data received from each server may be distributed.

[952] The scope of the disclosure includes software or machine-executable commands (e.g., an operating system, an application, firmware, a program, etc.) for enabling operations according to the methods of various embodiments to be executed on an apparatus or a computer, a non-transitory computer-readable medium having such software or commands stored thereon and executable on the apparatus or the computer.

Industrial Applicability

[953] The embodiments of the present disclosure may be used to encode or decode an image.

Date Reçue/Date Received 2023-12-29

CLAIMS

1. An image decoding method performed by an image decoding device, the method comprising:

determining a residual sample of a current block; and

resetting a value of the residual sample based on a color space transform being applied,

wherein the resetting of the value of the residual sample is performed based on a half value of a chroma residual sample value.

2. The method of claim 1, wherein the half value of the chroma residual sample value is derived based on right-shift operation for the chroma residual sample value.

3. The method of claim 1, wherein the resetting of the value of the residual sample is performed based on a luma component residual sample value and the half value of the chroma residual sample value.

4. The method of claim 1, wherein the resetting of the value of the residual sample is performed based on a sum of a luma component residual sample value and the half value of the chroma residual sample value.

5. The method of claim 1, wherein the resetting of the value of the residual sample is performed based on a sum of a half value of a chroma Cb component residual sample value and a luma component residual sample value.

6. The method of claim 1, wherein a value of a luma component residual sample of the residual sample is reset based on adding a half value of a chroma Cb component residual sample to the value of the luma component residual sample.

7. The method of claim 6, wherein the half value of the chroma Cb component residual sample added to the value of the luma component residual sample is determined based on subtracting, from a value of the chroma Cb component residual sample, a value that is obtained based on right-shift operation for the value of the chroma Cb component residual sample.

8. The method of claim 1, wherein a value of a chroma Cb component residual sample of the residual sample is reset based on subtracting, from a luma component residual sample value, a half value of the chroma Cb component residual sample and a half value of a chroma Cr component residual sample.

Date Reçue/Date Received 2023-12-29

9. The method of claim 8, wherein the half value of the chroma Cb component residual sample is determined based on a value that is obtained by applying right-shift operation to the value of the chroma Cb component residual sample, and

the half value of the chroma Cr component residual sample is determined based on a value that is obtained based on right-shift operation for a value of the chroma Cr component residual sample.

10. The method of claim 1, wherein a value of a chroma Cr component residual sample of the residual sample is reset based on subtracting a half value of a chroma Cb component residual sample from a luma component residual sample value and adding a half value of the chroma Cr component residual sample thereto.

11. The method of claim 10, wherein the half value of the chroma Cb component residual sample is determined based on a value that is obtained based on right-shift operation for a value of the chroma Cb component residual sample, and

the half value of the chroma Cr component residual sample is determined based on subtracting, from the value of the chroma Cr component residual sample, a value that is obtained based on right-shift operation for the value of the chroma Cr component residual sample.

12. The method of claim 1, wherein the determining of the residual sample of the current block comprises:

determining a prediction mode of the current block based on prediction information obtained from a bitstream;

obtaining, based on the prediction mode of the current block, an information for the color space transform for the residual sample of the current block;

determining quantization parameter of the current block based on the information; and

determining transform coefficient of the current block based on the quantization parameter,

wherein the determining of the quantization parameter is performed based on clipping for the quantization parameter such that a value of the quantization parameter has a value in a predetermined range.

13. An image decoding device comprising a memory and at least one processor,

wherein the at least one processor is configured to

determine a residual sample of a current block, and

reset a value of the residual sample based on a color space transform being applied, and

Date Reçue/Date Received 2023-12-29

the processor is configured to reset the value of the residual sample based on a half value of a chroma residual sample value.

14. An image encoding method performed by an image encoding device, the method comprising:

determining a residual sample of a current block; and

resetting a value of the residual sample based on a color space transform being applied,

wherein the resetting of the value of the residual sample is performed based on a half value of a chroma residual sample value.

15. A method for transmitting a bitstream generated by the image encoding method of claim 14.

Date Reçue/Date Received 2023-12-29

FIG. 1

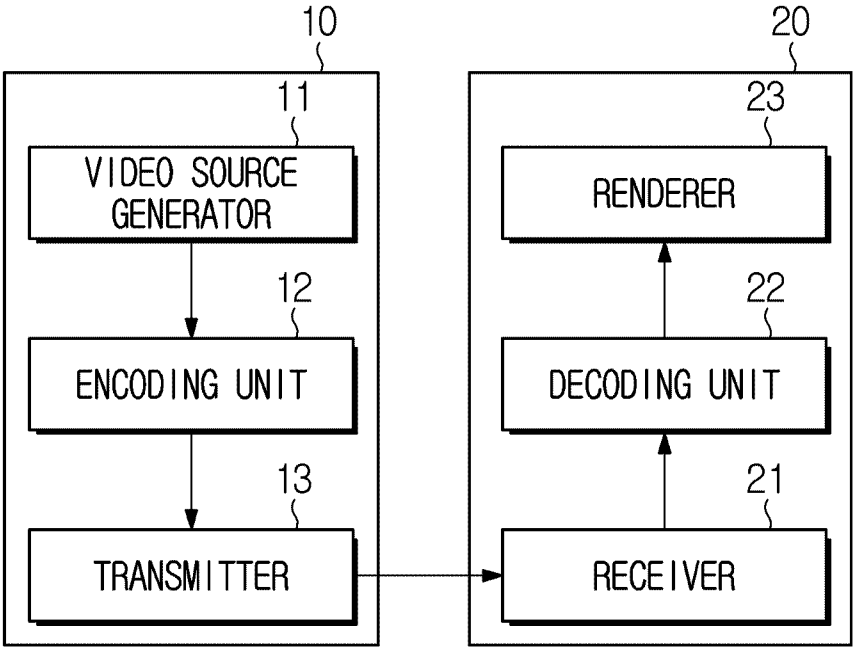

10
11
VIDEO SOURCE GENERATOR
12
ENCODING UNIT
13
TRANSMITTER
20
23
RENDERER
22
DECODING UNIT
21
RECEIVER

FIG. 2

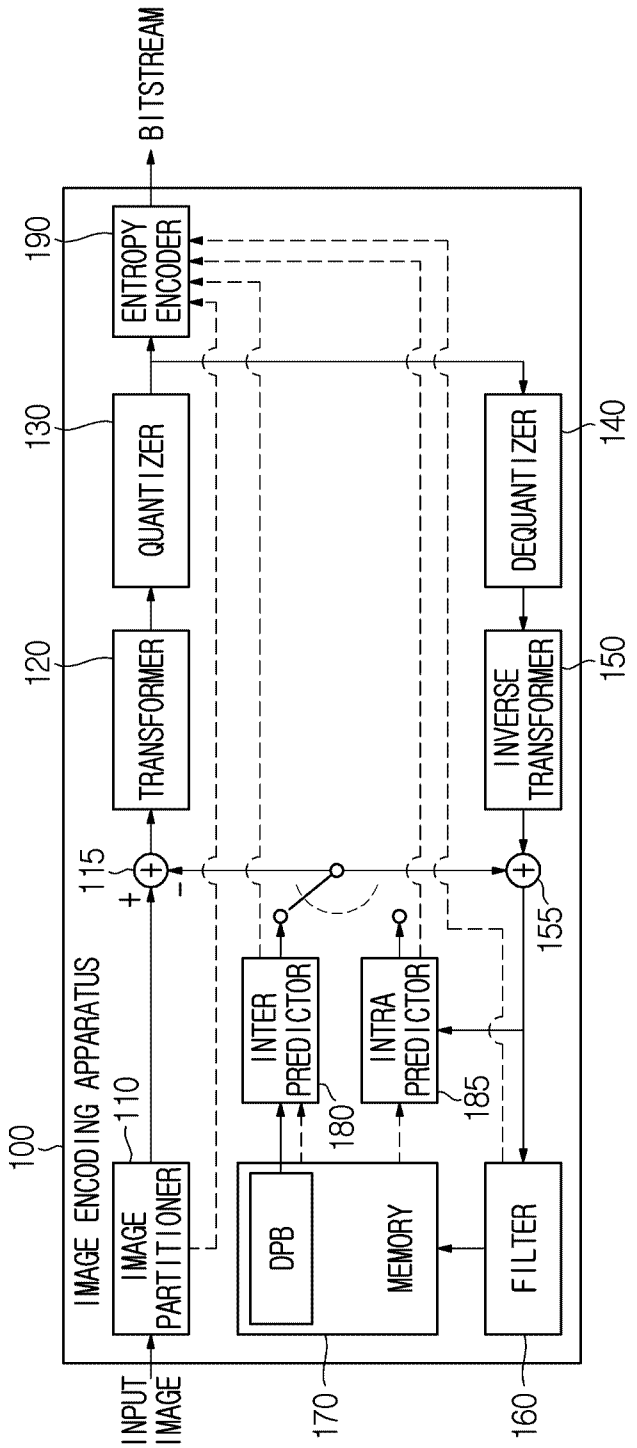

BITSTREAM
ENTROPY ENCODER
190
QUANTIZER
130
DEQUANTIZER
140
TRANSFORMER
120
INVERSE TRANSFORMER
150
115
155
INTER PREDICTOR
180
INTRA PREDICTOR
185
IMAGE ENCODING APPARATUS
100
IMAGE PARTITIONER
110
DPB
MEMORY
170
FILTER
160
INPUT IMAGE

FIG. 3

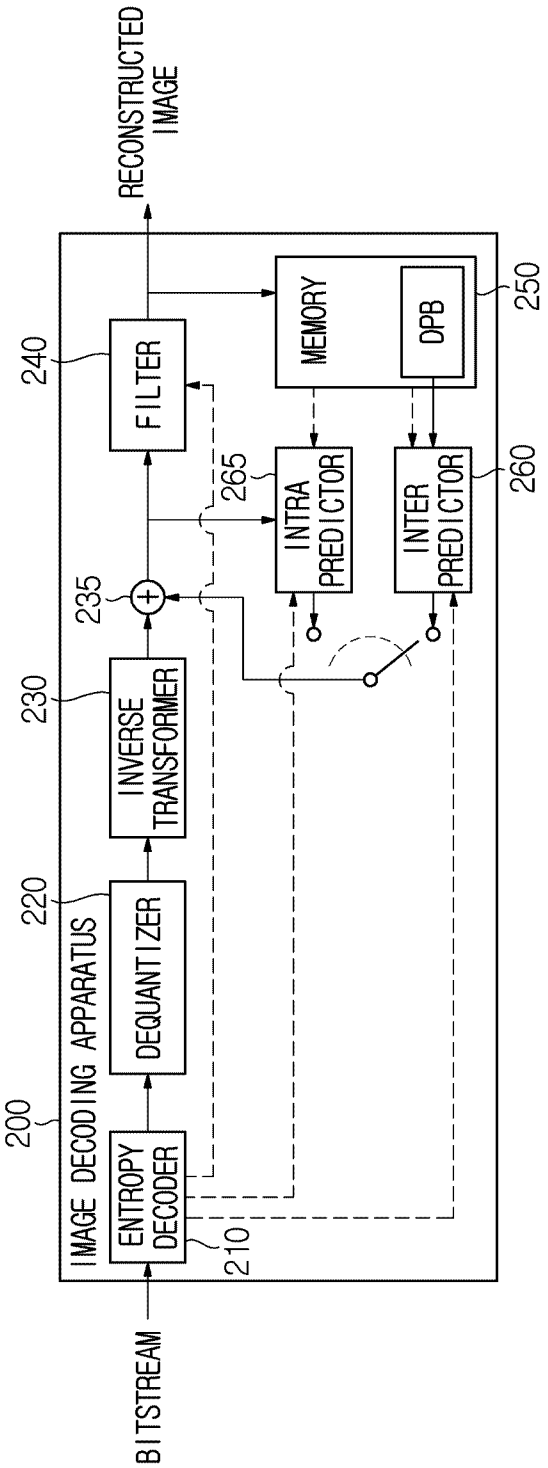

RECONSTRUCTED IMAGE
IMAGE DECODING APPARATUS
200
ENTROPY DECODER
210
DEQUANTIZER
220
INVERSE TRANSFORMER
230
235
FILTER
240
INTRA PREDICTOR
265
INTER PREDICTOR
260
MEMORY
DPB
250
BITSTREAM

FIG. 4

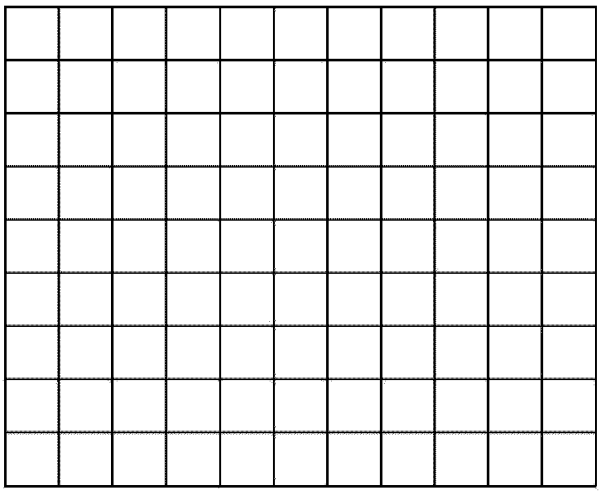


FIG. 5

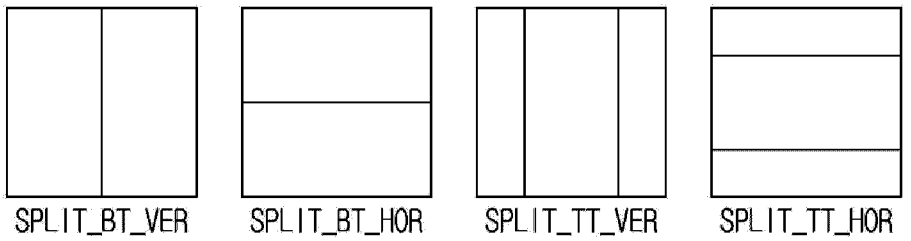


FIG. 6

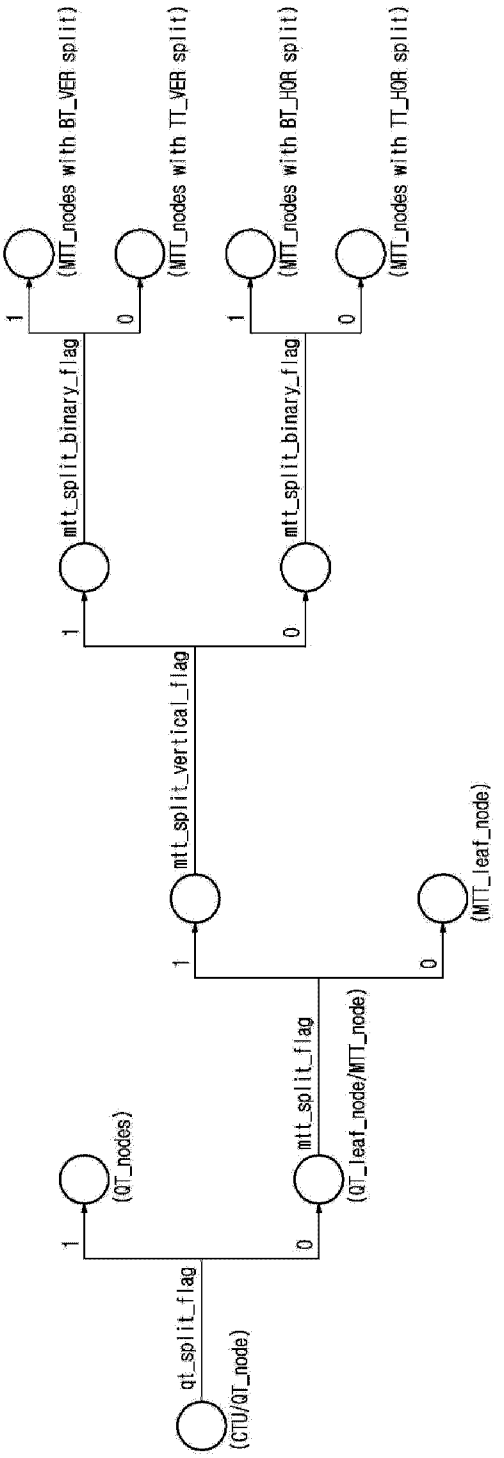

(CTU/QT_node)
qt_split_flag
1
(QT_nodes)
0
mtt_split_flag
(QT_leaf_node/MTT_node)
1
mtt_split_vertical_flag
0
(MTT_leaf_node)
1
mtt_split_binary_flag
(MTT_nodes with BT_VER split)
0
(MTT_nodes with TT_VER split)
0
mtt_split_binary_flag
(MTT_nodes with BT_HOR split)
1
(MTT_nodes with TT_HOR split)

FIG. 7

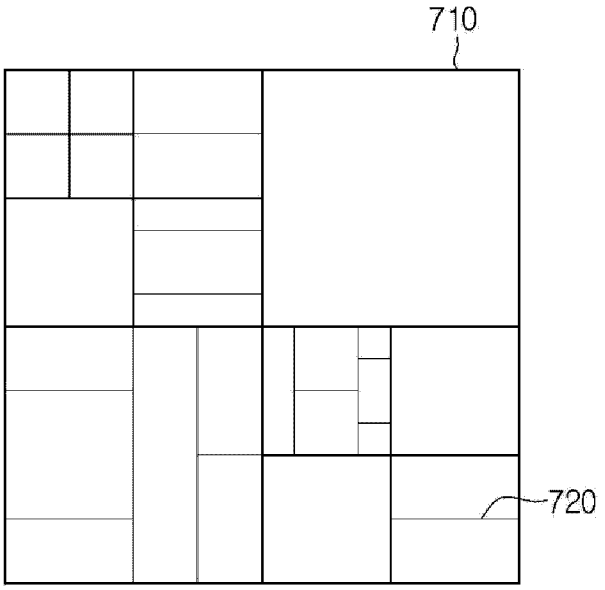


FIG. 8

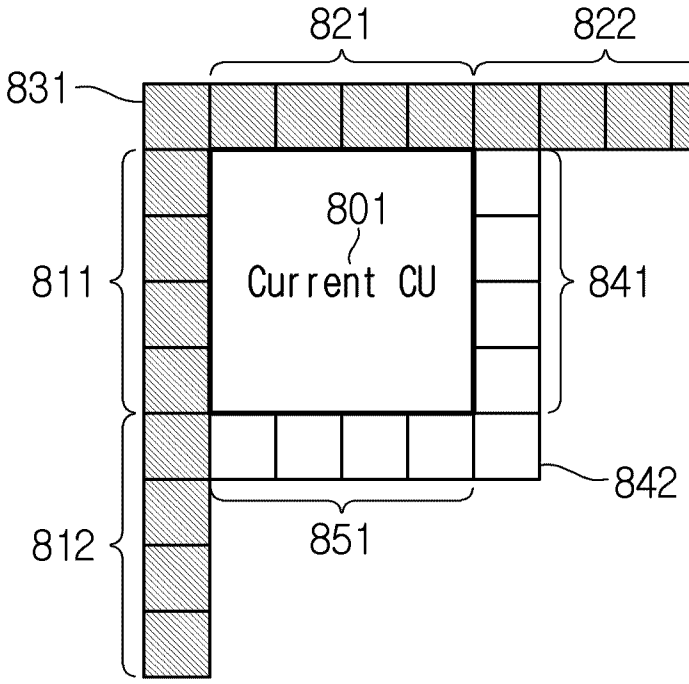
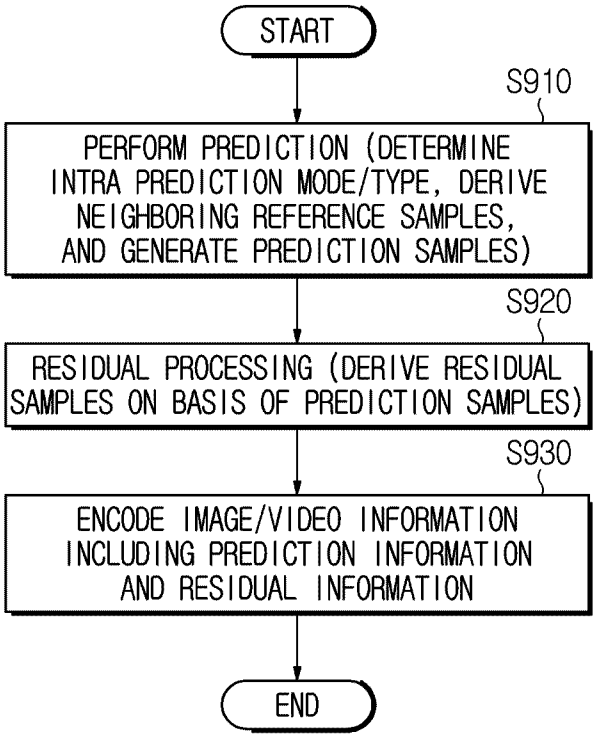


FIG. 9



Date Reçue/Date Received 2023-12-29

FIG. 10

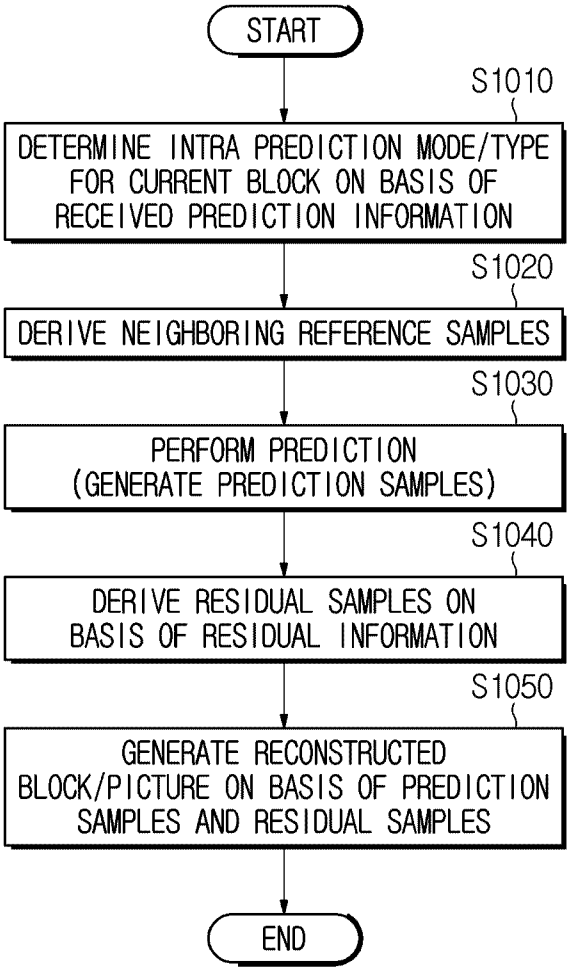
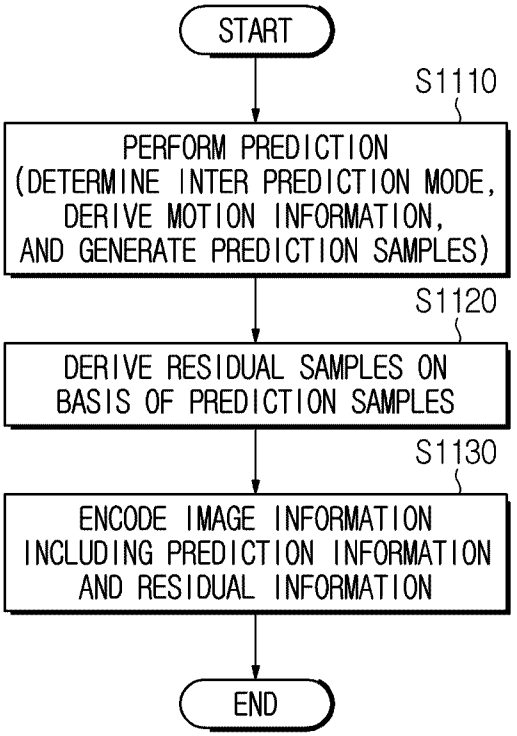

START
S1010
DETERMINE INTRA PREDICTION MODE/TYPE FOR CURRENT BLOCK ON BASIS OF RECEIVED PREDICTION INFORMATION
S1020
DERIVE NEIGHBORING REFERENCE SAMPLES
S1030
PERFORM PREDICTION (GENERATE PREDICTION SAMPLES)
S1040
DERIVE RESIDUAL SAMPLES ON BASIS OF RESIDUAL INFORMATION
S1050
GENERATE RECONSTRUCTED BLOCK/PICTURE ON BASIS OF PREDICTION SAMPLES AND RESIDUAL SAMPLES
END

FIG. 11



Date Reçue/Date Received 2023-12-29

FIG. 12

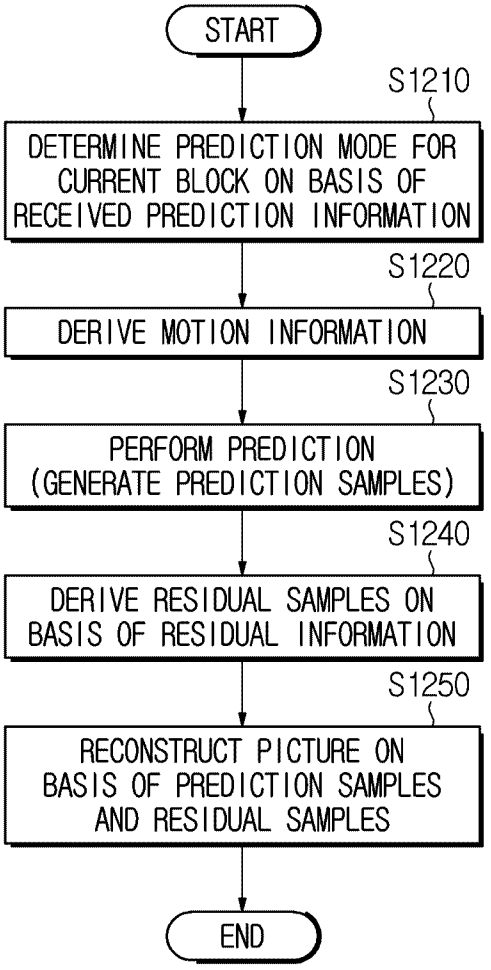

START
S1210
DETERMINE PREDICTION MODE FOR CURRENT BLOCK ON BASIS OF RECEIVED PREDICTION INFORMATION
S1220
DERIVE MOTION INFORMATION
S1230
PERFORM PREDICTION (GENERATE PREDICTION SAMPLES)
S1240
DERIVE RESIDUAL SAMPLES ON BASIS OF RESIDUAL INFORMATION
S1250
RECONSTRUCT PICTURE ON BASIS OF PREDICTION SAMPLES AND RESIDUAL SAMPLES
END

FIG. 13

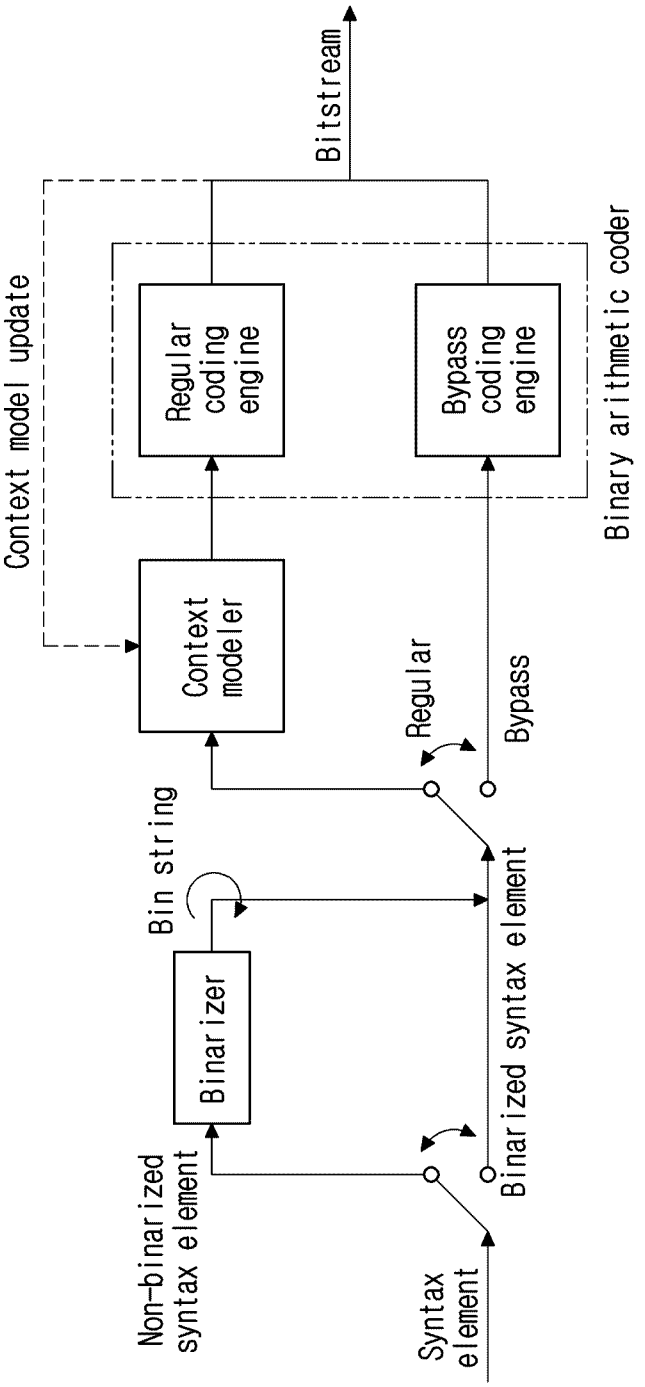

Bitstream
Context model update
Regular coding engine
Bypass coding engine
Binary arithmetic coder
Context modeler
Regular
Bypass
Bin string
Binarizer
Binarized syntax element
Non-binarized syntax element
Syntax element

FIG. 14

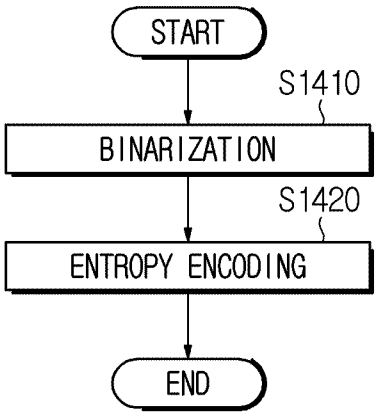

START
S1410
BINARIZATION
S1420
ENTROPY ENCODING
END

FIG. 15

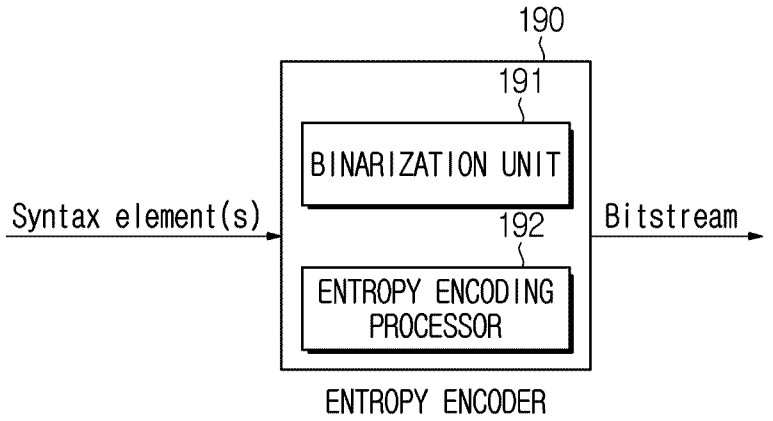

190
191
BINARIZATION UNIT
Syntax element(s)
192
Bitstream
ENTROPY ENCODING PROCESSOR
ENTROPY ENCODER

FIG. 16

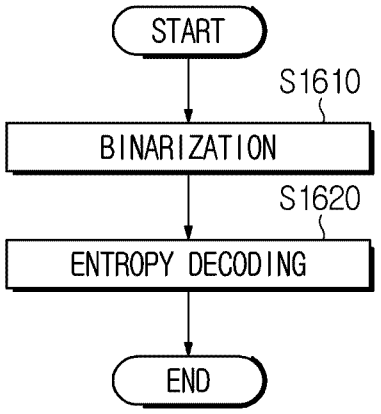

START
S1610
BINARIZATION
S1620
ENTROPY DECODING
END

FIG. 17

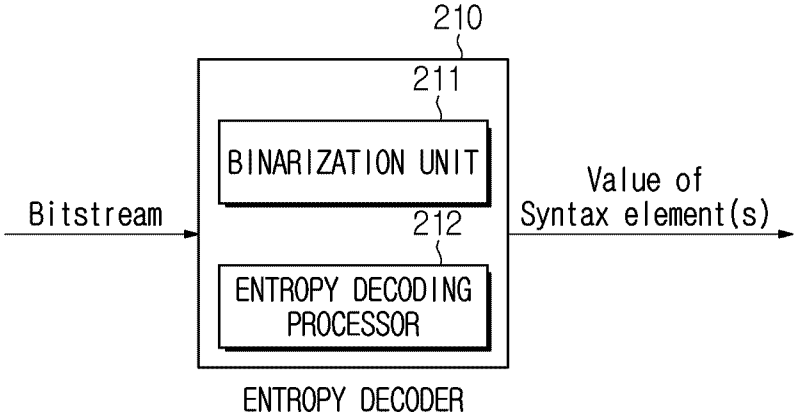

210
211
BINARIZATION UNIT
212
ENTROPY DECODING PROCESSOR
Bitstream
Value of Syntax element(s)
ENTROPY DECODER

FIG. 18

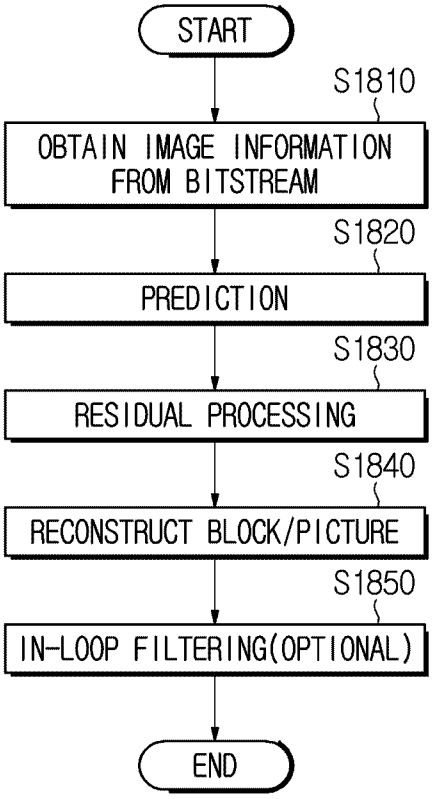

START
S1810
OBTAIN IMAGE INFORMATION FROM BITSTREAM
S1820
PREDICTION
S1830
RESIDUAL PROCESSING
S1840
RECONSTRUCT BLOCK/PICTURE
S1850
IN-LOOP FILTERING(OPTIONAL)
END

FIG. 19

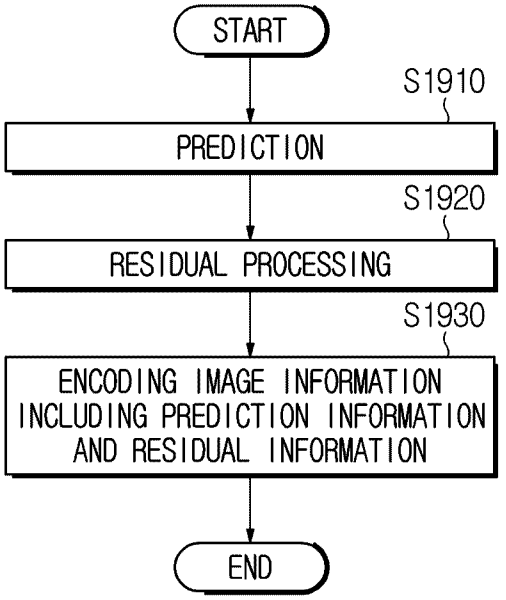

START
S1910
PREDICTION
S1920
RESIDUAL PROCESSING
S1930
ENCODING IMAGE INFORMATION INCLUDING PREDICTION INFORMATION AND RESIDUAL INFORMATION
END

FIG. 20

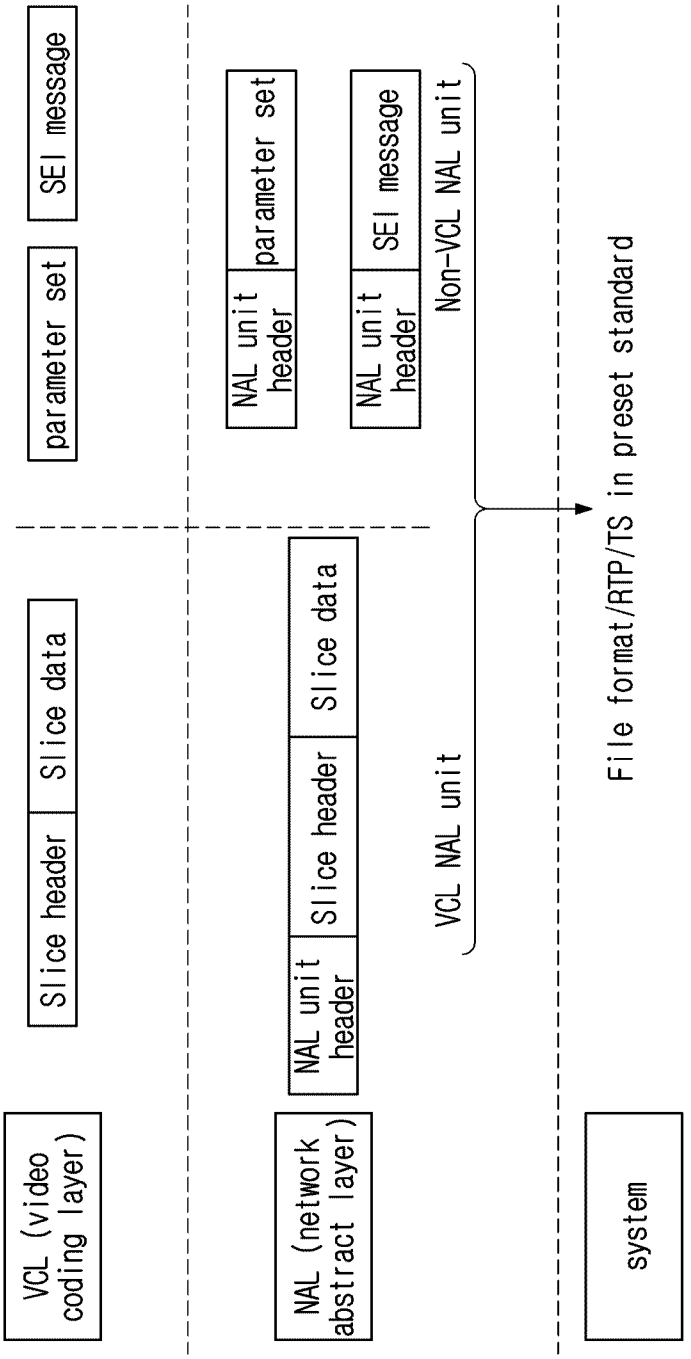

VCL (video coding layer)
NAL (network abstract layer)
system
Slice header
Slice data
NAL unit header
Slice header
Slice data
parameter set
SEI message
NAL unit header
parameter set
NAL unit header
SEI message
VCL NAL unit
Non-VCL NAL unit
File format/RTP/TS in preset standard

FIG. 21

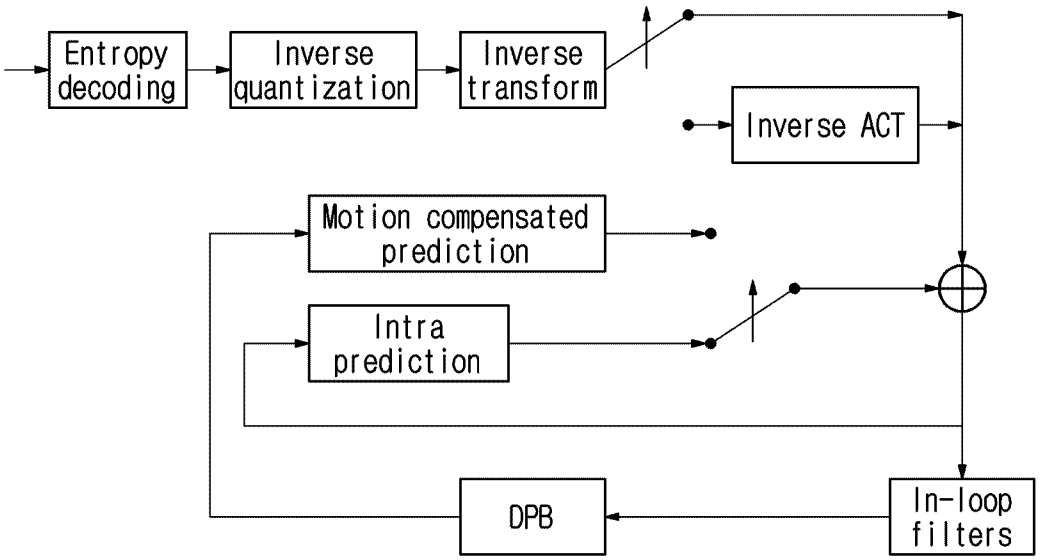


FIG. 22

2210

seq_parameter_set_rbsp( ) {	Descriptor
sps_decoding_parameter_set_id	u(4)
.....	
if(chroma_format_idc = = 3) {	
sps_palette_enabled_flag	u(1)
sps_act_enabled_flag	u(1)
}	

FIG. 23

coding_unit(x0, y0, cbWidth, cbHeight, cqtDepth, treeType, modeType) {	**Descriptor**
chType = treeType = = DUAL_TREE_CHROMA? 1 : 0	
if(slice_type != I || sps_ibc_enabled_flag || sps_palette_enabled_flag) {	
if(treeType != DUAL_TREE_CHROMA && !(((cbWidth = = 4 && cbHeight = = 4) || modeType = = MODE_TYPE_INTRA) && !sps_ibc_enabled_flag))	
cu_skip_flag[x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] = = 0 && slice_type != I && !(cbWidth = = 4 && cbHeight = = 4) && modeType = = MODE_TYPE_ALL)	
pred_mode_flag	ae(v)
if(((slice_type = = I && cu_skip_flag[x0][y0] = =0) || (slice_type != I && (CuPredMode[chType][x0][y0] != MODE_INTRA || (cbWidth = = 4 && cbHeight = = 4 && cu_skip_flag[x0][y0] = = 0)))) && cbWidth <= 64 && cbHeight <= 64 && modeType != MODE_TYPE_INTER && sps_ibc_enabled_flag && treeType != DUAL_TREE_CHROMA)	
pred_mode_ibc_flag	ae(v)
if(CuPredMode[x0][y0] = = MODE_INTRA && sps_palette_enabled_flag && cbWidth <= 64 && cbHeight <= 64 && cu_skip_flag[x0][y0] = = 0 && modeType != MODE_TYPE_INTER)	
pred_mode_plt_flag	ae(v)
}	
if(CuPredMode[chType][x0][y0] = = MODE_INTRA && sps_act_enabled_flag && treeType = = SINGLE_TREE)	
2310 **cu_act_enabled_flag**	ae(v)

FIG. 24

if(CuPredMode[chType][x0][y0] == MODE_INTRA ||	
CuPredMode[chType][x0][y0] == MODE_PLT) {	
if(treeType == SINGLE_TREE || treeType == DUAL_TREE_LUMA) {	
if(pred_mode_plt_flag) {	
if(treeType == DUAL_TREE_LUMA)	
palette_coding(x0, y0, cbWidth, cbHeight, 0, 1)	
else /* SINGLE_TREE */	
palette_coding(x0, y0, cbWidth, cbHeight, 0, 3)	
} else {	
if(sps_bdpcm_enabled_flag &&	
cbWidth <= MaxTsSize && cbHeight <= MaxTsSize)	
intra_bdpcm_luma_flag	ae(v)
if(intra_bdpcm_luma_flag)	
intra_bdpcm_luma_dir_flag	ae(v)
else {	
if(sps_mip_enabled_flag)	
intra_mip_flag[x0][y0]	ae(v)
if(intra_mip_flag[x0][y0]) {	
intra_mip_transposed[x0][y0]	ae(v)
intra_mip_mode[x0][y0]	ae(v)
} else {	
if(sps_mrl_enabled_flag && ((y0 % CtbSizeY) > 0))	
intra_luma_ref_idx[x0][y0]	ae(v)
if(sps_isp_enabled_flag && intra_luma_ref_idx[x0][y0] == 0 &&	
(cbWidth <= MaxTbSizeY && cbHeight <= MaxTbSizeY) &&	
(cbWidth * cbHeight > MinTbSizeY * MinTbSizeY) && !cu_act_enabled_flag)	
intra_subpartitions_mode_flag[x0][y0]	ae(v)
if(intra_subpartitions_mode_flag[x0][y0] == 1)	
intra_subpartitions_split_flag[x0][y0]	ae(v)
if(intra_luma_ref_idx[x0][y0] == 0)	
intra_luma_mpm_flag[x0][y0]	ae(v)
if(intra_luma_mpm_flag[x0][y0]) {	
if(intra_luma_ref_idx[x0][y0] == 0)	
intra_luma_not_planar_flag[x0][y0]	ae(v)
if(intra_luma_not_planar_flag[x0][y0])	
intra_luma_mpm_idx[x0][y0]	ae(v)
} else	
intra_luma_mpm_remainder[x0][y0]	ae(v)
}	
}	
}	
}	

FIG. 25

if((treeType == SINGLE_TREE || treeType == DUAL_TREE_CHROMA) && ChromaArrayType != 0) {	
if(pred_mode_plt_flag && treeType == DUAL_TREE_CHROMA)	
palette_coding(x0, y0, cbWidth / SubWidthC, cbHeight / SubHeightC, 1, 2)	
else {	
if(!cu_act_enabled_flag) {	
if(cbWidth <= MaxTsSize && cbHeight <= MaxTsSize && sps_bdpcm_chroma_enabled_flag) {	
intra_bdpcm_chroma_flag	ae(v)
if(intra_bdpcm_chroma_flag)	
intra_bdpcm_chroma_dir_flag	ae(v)
} else {	
if(CclmEnabled)	
cclm_mode_flag	ae(v)
if(cclm_mode_flag)	
cclm_mode_idx	ae(v)
else	
intra_chroma_pred_mode	ae(v)
}	
}	
}	
}	
} else if(treeType != DUAL_TREE_CHROMA) { /* MODE_INTER or MODE_IBC */	
if(cu_skip_flag[x0][y0] == 0)	
general_merge_flag[x0][y0]	ae(v)
if(general_merge_flag[x0][y0])	
merge_data(x0, y0, cbWidth, cbHeight, chType)	
else if(CuPredMode[chType][x0][y0] == MODE_IBC) {	
mvd_coding(x0, y0, 0, 0)	
if(MaxNumIbcMergeCand > 1)	
mvp_l0_flag[x0][y0]	ae(v)
if(sps_amvr_enabled_flag && (MvdL0[x0][y0][0] != 0 || MvdL0[x0][y0][1] != 0))	
amvr_precision_idx[x0][y0]	ae(v)

FIG. 26

} else {	
if(slice_type == B)	
inter_pred_idc[x0][y0]	ae(v)
if(sps_affine_enabled_flag && cbWidth >= 16 && cbHeight >= 16) {	
inter_affine_flag[x0][y0]	ae(v)
if(sps_affine_type_flag && inter_affine_flag[x0][y0])	
cu_affine_type_flag[x0][y0]	ae(v)
}	
if(sps_smvd_enabled_flag && !mvd_l1_zero_flag && inter_pred_idc[x0][y0] == PRED_BI && !inter_affine_flag[x0][y0] && RefIdxSymL0 > −1 && RefIdxSymL1 > −1)	
sym_mvd_flag[x0][y0]	ae(v)
if(inter_pred_idc[x0][y0] != PRED_L1) {	
if(NumRefIdxActive[0] > 1 && !sym_mvd_flag[x0][y0])	
ref_idx_l0[x0][y0]	ae(v)
mvd_coding(x0, y0, 0, 0)	
if(MotionModelIdc[x0][y0] > 0)	
mvd_coding(x0, y0, 0, 1)	
if(MotionModelIdc[x0][y0] > 1)	
mvd_coding(x0, y0, 0, 2)	
mvp_l0_flag[x0][y0]	ae(v)
} else {	
MvdL0[x0][y0][0] = 0	
MvdL0[x0][y0][1] = 0	
}	
if(inter_pred_idc[x0][y0] != PRED_L0) {	
if(NumRefIdxActive[1] > 1 && !sym_mvd_flag[x0][y0])	
ref_idx_l1[x0][y0]	ae(v)
if(mvd_l1_zero_flag && inter_pred_idc[x0][y0] == PRED_BI) {	
MvdL1[x0][y0][0] = 0	
MvdL1[x0][y0][1] = 0	
MvdCpL1[x0][y0][0][0] = 0	
MvdCpL1[x0][y0][0][1] = 0	
MvdCpL1[x0][y0][1][0] = 0	
MvdCpL1[x0][y0][1][1] = 0	
MvdCpL1[x0][y0][2][0] = 0	
MvdCpL1[x0][y0][2][1] = 0	

FIG. 27

} else {	
if(sym_mvd_flag[x0][y0]) {	
MvdL1[x0][y0][0] = −MvdL0[x0][y0][0]	
MvdL1[x0][y0][1] = −MvdL0[x0][y0][1]	
} else	
mvd_coding(x0, y0, 1, 0)	
if(MotionModelIdc[x0][y0] > 0)	
mvd_coding(x0, y0, 1, 1)	
if(MotionModelIdc[x0][y0] > 1)	
mvd_coding(x0, y0, 1, 2)	
mvp_l1_flag[x0][y0]	ae(v)
}	
} else {	
MvdL1[x0][y0][0] = 0	
MvdL1[x0][y0][1] = 0	
}	
if((sps_amvr_enabled_flag && inter_affine_flag[x0][y0] == 0 && (MvdL0[x0][y0][0] != 0 || MvdL0[x0][y0][1] != 0 || MvdL1[x0][y0][0] != 0 || MvdL1[x0][y0][1] != 0)) || (sps_affine_amvr_enabled_flag && inter_affine_flag[x0][y0] == 1 && (MvdCpL0[x0][y0][0][0] != 0 || MvdCpL0[x0][y0][0][1] != 0 || MvdCpL1[x0][y0][0][0] != 0 || MvdCpL1[x0][y0][0][1] != 0 || MvdCpL0[x0][y0][1][0] != 0 || MvdCpL0[x0][y0][1][1] != 0 || MvdCpL1[x0][y0][1][0] != 0 || MvdCpL1[x0][y0][1][1] != 0 || MvdCpL0[x0][y0][2][0] != 0 || MvdCpL0[x0][y0][2][1] != 0 || MvdCpL1[x0][y0][2][0] != 0 || MvdCpL1[x0][y0][2][1] != 0)) {	
amvr_flag[x0][y0]	ae(v)
if(amvr_flag[x0][y0])	
amvr_precision_idx[x0][y0]	ae(v)
}	
if(sps_bcw_enabled_flag && inter_pred_idc[x0][y0] == PRED_BI && luma_weight_l0_flag[ref_idx_l0 [x0][y0]] == 0 && luma_weight_l1_flag[ref_idx_l1 [x0][y0]] == 0 && chroma_weight_l0_flag[ref_idx_l0 [x0][y0]] == 0 && chroma_weight_l1_flag[ref_idx_l1 [x0][y0]] == 0 && cbWidth * cbHeight >= 256)	
bcw_idx[x0][y0]	ae(v)
}	

FIG. 28

if(CuPredMode[chType][x0][y0] != MODE_INTRA && !pred_mode_plt_flag && general_merge_flag[x0][y0] == 0)	
cu_cbf	ae(v)
if(cu_cbf) {	
if(CuPredMode[chType][x0][y0] == MODE_INTER && sps_sbt_enabled_flag && !ciip_flag[x0][y0] && !MergeTriangleFlag[x0][y0]) {	
if(cbWidth <= MaxTbSizeY && cbHeight <= MaxTbSizeY) {	
allowSbtVerH = cbWidth >= 8	
allowSbtVerQ = cbWidth >= 16	
allowSbtHorH = cbHeight >= 8	
allowSbtHorQ = cbHeight >= 16	
if(allowSbtVerH || allowSbtHorH || allowSbtVerQ || allowSbtHorQ)	
cu_sbt_flag	ae(v)
}	
if(cu_sbt_flag) {	
if((allowSbtVerH || allowSbtHorH) && (allowSbtVerQ || allowSbtHorQ))	
cu_sbt_quad_flag	ae(v)
if((cu_sbt_quad_flag && allowSbtVerQ && allowSbtHorQ) || (!cu_sbt_quad_flag && allowSbtVerH && allowSbtHorH))	
cu_sbt_horizontal_flag	ae(v)
cu_sbt_pos_flag	ae(v)
}	
}	

FIG. 29

2910

if(sps_act_enabled_flag && CuPredMode[chType][x0][y0] != MODE_INTRA && treeType == SINGLE_TREE)	
cu_act_enabled_flag	ae(v)
LfnstDcOnly = 1	
LfnstZeroOutSigCoeffFlag = 1	
MtsZeroOutSigCoeffFlag = 1	
transform_tree(x0, y0, cbWidth, cbHeight, treeType, chType)	
lfnstWidth = (treeType == DUAL_TREE_CHROMA) ? cbWidth / SubWidthC : ((IntraSubPartitionsSplitType == ISP_VER_SPLIT) ? cbWidth / NumIntraSubPartitions : cbWidth)	
lfnstHeight = (treeType == DUAL_TREE_CHROMA) ? cbHeight / SubHeightC : ((IntraSubPartitionsSplitType == ISP_HOR_SPLIT) ? cbHeight / NumIntraSubPartitions : cbHeight)	
if(Min(lfnstWidth, lfnstHeight) >= 4 && sps_lfnst_enabled_flag == 1 && CuPredMode[chType][x0][y0] == MODE_INTRA && (!intra_mip_flag[x0][y0] || Min(lfnstWidth, lfnstHeight) >= 16) && Max(cbWidth, cbHeight) <= MaxTbSizeY) {	
if((IntraSubPartitionsSplitType != ISP_NO_SPLIT || LfnstDcOnly == 0) && LfnstZeroOutSigCoeffFlag == 1)	
lfnst_idx	ae(v)
}	
if(treeType != DUAL_TREE_CHROMA && lfnst_idx == 0 && transform_skip_flag[x0][y0][0] == 0 && Max(cbWidth, cbHeight) <= 32 && IntraSubPartitionsSplit[x0][y0] == ISP_NO_SPLIT && cu_sbt_flag == 0 && MtsZeroOutSigCoeffFlag == 1 && tu_cbf_luma[x0][y0]) {	
if(((CuPredMode[chType][x0][y0] == MODE_INTER && sps_explicit_mts_inter_enabled_flag) || (CuPredMode[chType][x0][y0] == MODE_INTRA && sps_explicit_mts_intra_enabled_flag)))	
mts_idx	ae(v)
}	
}	

FIG. 30

coding_tree(x0, y0, cbWidth, cbHeight, qgOnY, qgOnC, cbSubdiv, cqtDepth, mttDepth, depthOffset, partIdx, treeTypeCurr, modeTypeCurr) {	**Descriptor**
if((allowSplitBtVer || allowSplitBtHor || allowSplitTtVer || allowSplitTtHor || allowSplitQt) && (x0 + cbWidth <= pps_pic_width_in_luma_samples) && (y0 + cbHeight <= pps_pic_height_in_luma_samples))	
split_cu_flag	ae(v)
if(pps_cu_qp_delta_enabled_flag && qgOnY && cbSubdiv <= CuQpDeltaSubdiv) {	
IsCuQpDeltaCoded = 0	
CuQpDeltaVal = 0	
CuQgTopLeftX = x0	
CuQgTopLeftY = y0	
}	
...	...

FIG. 31

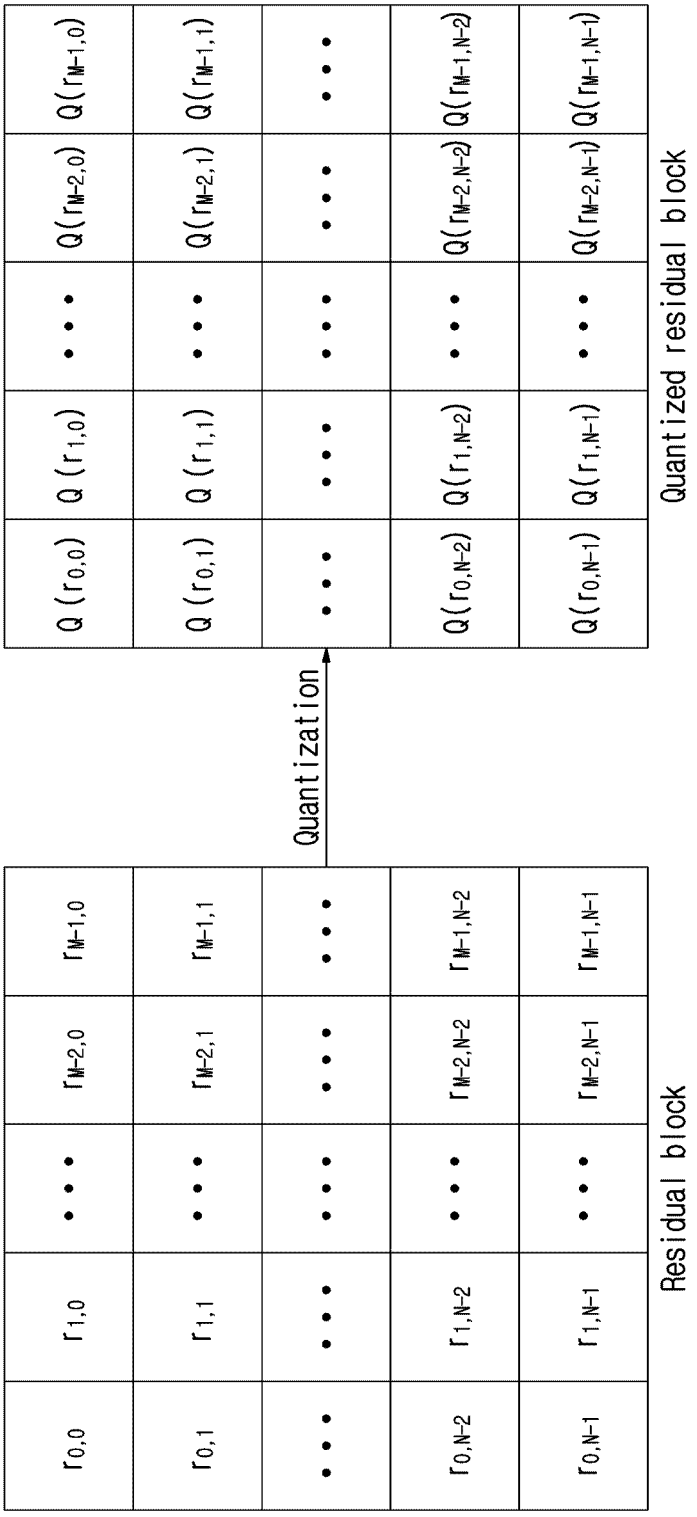


FIG. 32

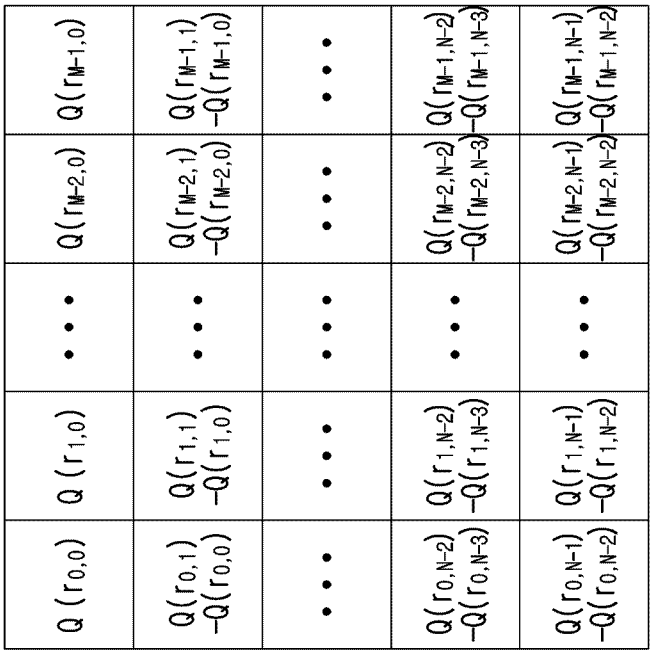

Q (r0,0)
Q(r1,0) −Q(r0,0)
• • •
Q(rM−2,0) −Q(rM−3,0)
Q(rM−1,0) −Q(rM−2,0)
Q (r0,1)
Q(r1,1) −Q(r0,1)
• • •
Q(rM−2,1) −Q(rM−3,1)
Q(rM−1,1) −Q(rM−2,1)
• • •
• • •
• • •
• • •
• • •
Q(r0,N−2)
Q(r1,N−2) −Q(r0,N−2)
• • •
Q(rM−2,N−2) −Q(rM−3,N−2)
Q(rM−1,N−2) −Q(rM−2,N−2)
Q(r0,N−1)
Q(r1,N−1) −Q(r0,N−1)
• • •
Q(rM−2,N−1) −Q(rM−3,N−1)
Q(rM−1,N−1) −Q(rM−2,N−1)
Horizontal BDPCM
Q (r0,0)
Q (r1,0)
• • •
Q(rM−2,0)
Q(rM−1,0)
Q(r0,1) −Q(r0,0)
Q(r1,1) −Q(r1,0)
• • •
Q(rM−2,1) −Q(rM−2,0)
Q(rM−1,1) −Q(rM−1,0)
• • •
• • •
• • •
• • •
• • •
Q(r0,N−2) −Q(r0,N−3)
Q(r1,N−2) −Q(r1,N−3)
• • •
Q(rM−2,N−2) −Q(rM−2,N−3)
Q(rM−1,N−2) −Q(rM−1,N−3)
Q(r0,N−1) −Q(r0,N−2)
Q(r1,N−1) −Q(r1,N−2)
• • •
Q(rM−2,N−1) −Q(rM−2,N−2)
Q(rM−1,N−1) −Q(rM−1,N−2)
Vertical BDPCM

FIG. 33

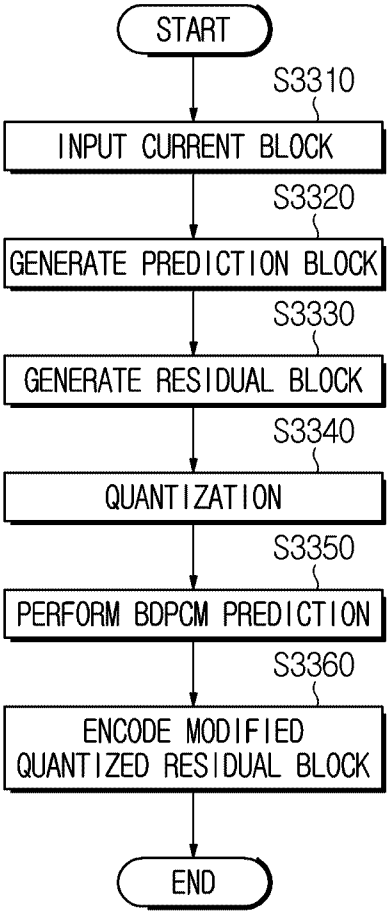
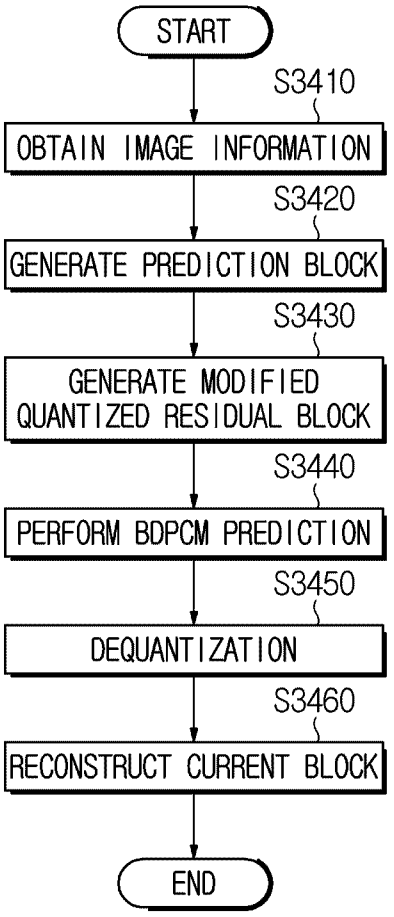

START
S3310
INPUT CURRENT BLOCK
S3320
GENERATE PREDICTION BLOCK
S3330
GENERATE RESIDUAL BLOCK
S3340
QUANTIZATION
S3350
PERFORM BDPCM PREDICTION
S3360
ENCODE MODIFIED QUANTIZED RESIDUAL BLOCK
END

FIG. 34



Date Reçue/Date Received 2023-12-29

FIG. 35

seq_parameter_set_rbsp( ) {	**Descriptor**
(…)	
sps_transform_skip_enabled_flag	u(1)
if(sps_transform_skip_enabled_flag)	
sps_bdpcm_enabled_flag	u(1)
(….)	
}	

FIG. 36

general_constraint_info( ) {	**Descriptor**
(…)	
no_transform_skip_constraint_flag	u(1)
no_bdpcm_constraint_flag	u(1)
(…)	
}	

FIG. 37

coding_unit(x0, y0, cbWidth, cbHeight, cqtDepth, treeType, modeType) {	Descriptor
chType = treeType = = DUAL_TREE_CHROMA? 1 : 0	
if(slice_type != I || sps_ibc_enabled_flag || sps_palette_enabled_flag) {	
if(treeType != DUAL_TREE_CHROMA && !((cbWidth = = 4 && cbHeight = = 4) || modeType = = MODE_TYPE_INTRA) && !sps_ibc_enabled_flag))	
cu_skip_flag[x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] = = 0 && slice_type != I && !(cbWidth = = 4 && cbHeight = = 4) && modeType = = MODE_TYPE_ALL)	
pred_mode_flag	ae(v)
if(((slice_type = = I && cu_skip_flag[x0][y0] = =0) || (slice_type != I && (CuPredMode[chType][x0][y0] != MODE_INTRA || (cbWidth = = 4 && cbHeight = = 4 && cu_skip_flag[x0][y0] = = 0)))) && cbWidth <= 64 && cbHeight <= 64 && modeType != MODE_TYPE_INTER && sps_ibc_enabled_flag && treeType != DUAL_TREE_CHROMA)	
pred_mode_ibc_flag	ae(v)
if((((slice_type = = I || (cbWidth = = 4 && cbHeight = = 4) || sps_ibc_enabled_flag) && CuPredMode[x0][y0] = = MODE_INTRA) || (slice_type != I && !(cbWidth = = 4 && cbHeight = = 4) && !sps_ibc_enabled_flag && CuPredMode[x0][y0] != MODE_INTRA)) && sps_palette_enabled_flag && cbWidth <= 64 && cbHeight <= 64 && && cu_skip_flag[x0][y0] = = 0 && modeType != MODE_INTER)	
pred_mode_plt_flag	ae(v)
}	
if(CuPredMode[chType][x0][y0] = = MODE_INTRA || CuPredMode[chType][x0][y0] = = MODE_PLT) {	
if(treeType = = SINGLE_TREE || treeType = = DUAL_TREE_LUMA) {	
if(pred_mode_plt_flag) {	
if(treeType = = DUAL_TREE_LUMA)	
palette_coding(x0, y0, cbWidth, cbHeight, 0, 1)	
else /* SINGLE_TREE */	
palette_coding(x0, y0, cbWidth, cbHeight, 0, 3)	
} else {	
if(sps_bdpcm_enabled_flag && cbWidth <= MaxTsSize && cbHeight <= MaxTsSize)	
intra_bdpcm_flag	ae(v)
if(intra_bdpcm_flag)	
intra_bdpcm_dir_flag	ae(v)
(…)	

FIG. 38

pic_parameter_set_rbsp( ) {	Descriptor
……	
pps_act_qp_offsets_present_flag	u(1)
pps_act_y_qp_offset_plus5	se(v)
pps_act_cb_qp_offset_plus5	se(v)
pps_act_cr_qp_offset_plus3	se(v)
…	
}	

FIG. 39

Coding_unit(x0, y0, cbWidth, cbHeight, cqtDepth, treeType, modeType) {	Descriptor
……	
if((treeType == SINGLE_TREE || treeType == DUAL_TREE_CHROMA) && ChromaArrayType != 0) {	
if(pred_mode_plt_flag && treeType == DUAL_TREE_CHROMA)	
palette_coding(x0, y0, cbWidth / SubWidthC, cbHeight / SubHeightC, 1, 2)	
else {	
~~if(!cu_act_enabled_flag) {~~	
if(cbWidth <= MaxTsSize && cbHeight <= MaxTsSize && sps_bdpcm_chroma_enabled_flag) {	
intra_bdpcm_chroma_flag	ae(v)
if(intra_bdpcm_chroma_flag)	
intra_bdpcm_chroma_dir_flag	ae(v)
} else if(!cu_act_enabled_flag) {	
if(CclmEnabled)	
cclm_mode_flag	ae(v)
if(cclm_mode_flag)	
cclm_mode_idx	ae(v)
else	
intra_chroma_pred_mode	ae(v)
}	
~~}~~	
}	
}	
} else if(treeType != DUAL_TREE_CHROMA) { /* MODE_INTER or MODE_IBC */	
…	

FIG. 40

Coding_unit(x0, y0, cbWidth, cbHeight, cqtDepth, treeType, modeType) {	Descriptor
……	
if((treeType == SINGLE_TREE || treeType == DUAL_TREE_CHROMA) && ChromaArrayType != 0) {	
if(pred_mode_plt_flag && treeType == DUAL_TREE_CHROMA)	
palette_coding(x0, y0, cbWidth / SubWidthC, cbHeight / SubHeightC, 1, 2)	
else {	
~~if(!cu_act_enabled_flag) {~~	
if(cbWidth <= MaxTsSize && cbHeight <= MaxTsSize && sps_bdpcm_chroma_enabled_flag) {	
intra_bdpcm_chroma_flag	ae(v)
if(intra_bdpcm_chroma_flag)	
intra_bdpcm_chroma_dir_flag	ae(v)
} else {	
if(CclmEnabled)	
cclm_mode_flag	ae(v)
if(cclm_mode_flag)	
cclm_mode_idx	ae(v)
else	
intra_chroma_pred_mode	ae(v)
}	
~~}~~	
}	
}	
} else if(treeType != DUAL_TREE_CHROMA) { /* MODE_INTER or MODE_IBC */	
…	

FIG. 41

pic_parameter_set_rbsp( ) {	Descriptor
……	
pps_act_qp_offsets_present_flag	u(1)
if (pps_act_qp_offsets_present_flag) {	
pps_act_y_qp_offset_plusX1	se(v)
pps_act_cb_qp_offset_plusX2	se(v)
pps_act_cr_qp_offset_plusX3	se(v)
pps_act_cbcr_qp_offset_plusX4	se(v)
}	
…	
}	
…	

FIG. 42

pic_parameter_set_rbsp( ) {	Descriptor
……	
pps_act_qp_offsets_present_flag	u(1)
if (pps_act_qp_offsets_present_flag) {	
pps_act_y_qp_offset_plusX1	se(v)
pps_act_cb_qp_offset_plusX2	se(v)
pps_act_cr_qp_offset_plusX3	se(v)
pps_act_cbcr_qp_offset_modeA_plusX4	se(v)
pps_act_cbcr_qp_offset_modeB_plusX5	se(v)
}	
…	
}	

FIG. 43

pic_parameter_set_rbsp( ) {	Descriptor
……	
pps_act_qp_offsets_present_flag	u(1)
if (pps_act_qp_offsets_present_flag) {	
pps_act_y_qp_offset_plusX1	se(v)
pps_act_cb_qp_offset_plusX2	se(v)
pps_act_cr_qp_offset_plusX3	se(v)
}	
…	
}	

FIG. 44

pic_parameter_set_rbsp( ) {	Descriptor
……	
pps_slice_act_qp_offsets_present_flag	u(1)
…	
…	

FIG. 45

slice_header_rbsp( ) {	Descriptor
…..	
if (pps_slice_act_qp_offsets_present_flag) {	
slice_act_y_qp_offset	se(v)
slice_act_cb_qp_offset	se(v)
slice_act_cr_qp_offset	se(v)
if(sps_joint_cbcr_enabled_flag)	
slice_act_cbcr_qp_offset	se(v)
…	
}	
}	

FIG. 46

slice_header_rbsp( ) {	Descriptor
.....	
if (pps_slice_act_qp_offsets_present_flag) {	
slice_act_y_qp_offset	se(v)
slice_act_cb_qp_offset	se(v)
slice_act_cr_qp_offset	se(v)
if(sps_joint_cbcr_enabled_flag) {	
slice_act_cbcr_qp_offset_modeA	se(v)
slice_act_cbcr_qp_offset_modeB	se(v)
}	
...	
}	
}	

FIG. 47

slice_header_rbsp( ) {	Descriptor
.....	
if (pps_slice_act_qp_offsets_present_flag) {	
slice_act_y_qp_offset	se(v)
slice_act_cb_qp_offset	se(v)
slice_act_cr_qp_offset	se(v)
}	
...	
}	

FIG. 48

pic_parameter_set_rbsp( ) {	Descriptor
pps_pic_slice_act_qp_offsets_present_flag	u(1)

FIG. 49

picture_header_rbsp( ) {	Descriptor
if(pps_pic_slice_act_qp_offsets_present_flag) {	
pic_act_qp_offsets_present_flag	u(1)
if(pic_act_qp_offsets_present_flag) {	
slice_act_y_qp_offset	se(v)
slice_act_cb_qp_offset	se(v)
slice_act_cr_qp_offset	se(v)
}	
}	
}	

FIG. 50

slice_header_rbsp( ) {	Descriptor
if(pps_pic_slice_act_qp_offsets_present_flag && !pic_act_qp_offsets_present_flag) {	
slice_act_y_qp_offset	se(v)
slice_act_cb_qp_offset	se(v)
slice_act_cr_qp_offset	se(v)
}	
}	

FIG. 51

pic_parameter_set_rbsp( ) {	**Descriptor**
…	
pps_cb_qp_offset	se(v)
pps_cr_qp_offset	se(v)
pps_joint_cbcr_qp_offset_present_flag	u(1)
if(pps_joint_cbcr_qp_offset_present_flag)	
pps_joint_cbcr_qp_offset_value	se(v)
pps_slice_chroma_qp_offsets_present_flag	u(1)
pps_cu_chroma_qp_offset_list_enabled_flag	u(1)
if(pps_cu_chroma_qp_offset_list_enabled_flag) {	
chroma_qp_offset_list_len_minus1	ue(v)
for(i = 0; i <= chroma_qp_offset_list_len_minus1; i++) {	
cb_qp_offset_list[i]	se(v)
cr_qp_offset_list[i]	se(v)
if(pps_joint_cbcr_qp_offset_present_flag)	
joint_cbcr_qp_offset_list[i]	se(v)
}	
}	
pps_act_qp_offsets_present_flag	u(1)
if(pps_act_qp_offsets_present_flag)	
pps_act_y_qp_offset	se(v)
pps_act_cb_qp_offset	se(v)
pps_act_cr_qp_offset	se(v)
if(pps_joint_cbcr_qp_offset_present_flag)	
pps_act_cbcr_qp_offset	se(v)
pps_cu_act_qp_offset_list_enabled_flag	u(1)
if(pps_cu_act_qp_offset_list_enabled_flag) {	
for(i = 0; i <= chroma_qp_offset_list_len_minus1; i++) {	
act_y_qp_offset_list[i]	se(v)
act_cb_qp_offset_list[i]	se(v)
act_cr_qp_offset_list[i]	se(v)
if(pps_joint_cbcr_qp_offset_present_flag)	
act_cbcr_qp_offset_list[i]	se(v)
}	
}	
}	
…	
}	

FIG. 52

seq_parameter_set_rbsp( ) {	**Descriptor**
…	
sps_act_reversible_conversion	**u(1)**

FIG. 53

seq_parameter_set_rbsp( ) {	Descriptor
…	
sps_act_shift_flag	u(1)
…	

FIG. 54

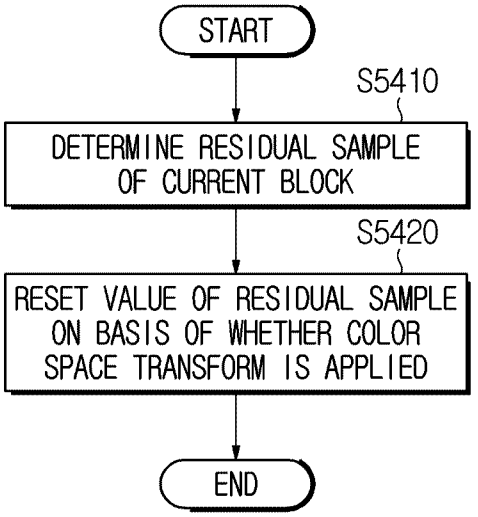


FIG. 55

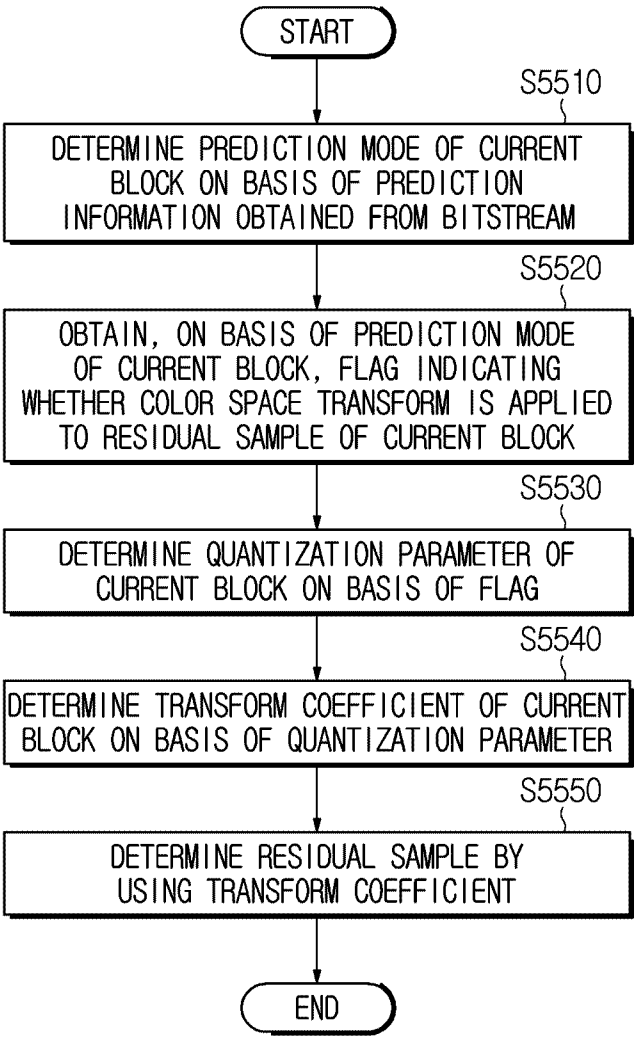

START
S5510
DETERMINE PREDICTION MODE OF CURRENT BLOCK ON BASIS OF PREDICTION INFORMATION OBTAINED FROM BITSTREAM
S5520
OBTAIN, ON BASIS OF PREDICTION MODE OF CURRENT BLOCK, FLAG INDICATING WHETHER COLOR SPACE TRANSFORM IS APPLIED TO RESIDUAL SAMPLE OF CURRENT BLOCK
S5530
DETERMINE QUANTIZATION PARAMETER OF CURRENT BLOCK ON BASIS OF FLAG
S5540
DETERMINE TRANSFORM COEFFICIENT OF CURRENT BLOCK ON BASIS OF QUANTIZATION PARAMETER
S5550
DETERMINE RESIDUAL SAMPLE BY USING TRANSFORM COEFFICIENT
END

FIG. 56

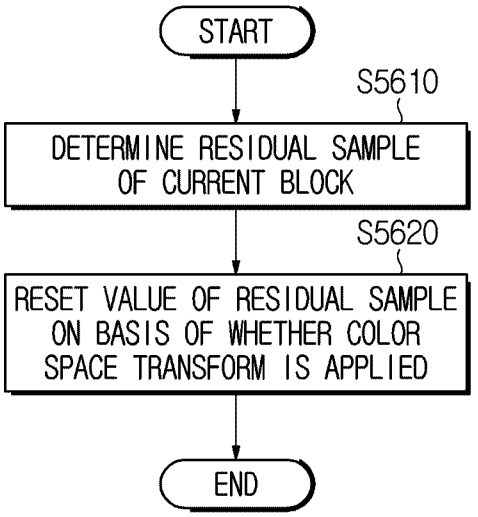

START
S5610
DETERMINE RESIDUAL SAMPLE OF CURRENT BLOCK
S5620
RESET VALUE OF RESIDUAL SAMPLE ON BASIS OF WHETHER COLOR SPACE TRANSFORM IS APPLIED
END

FIG. 57

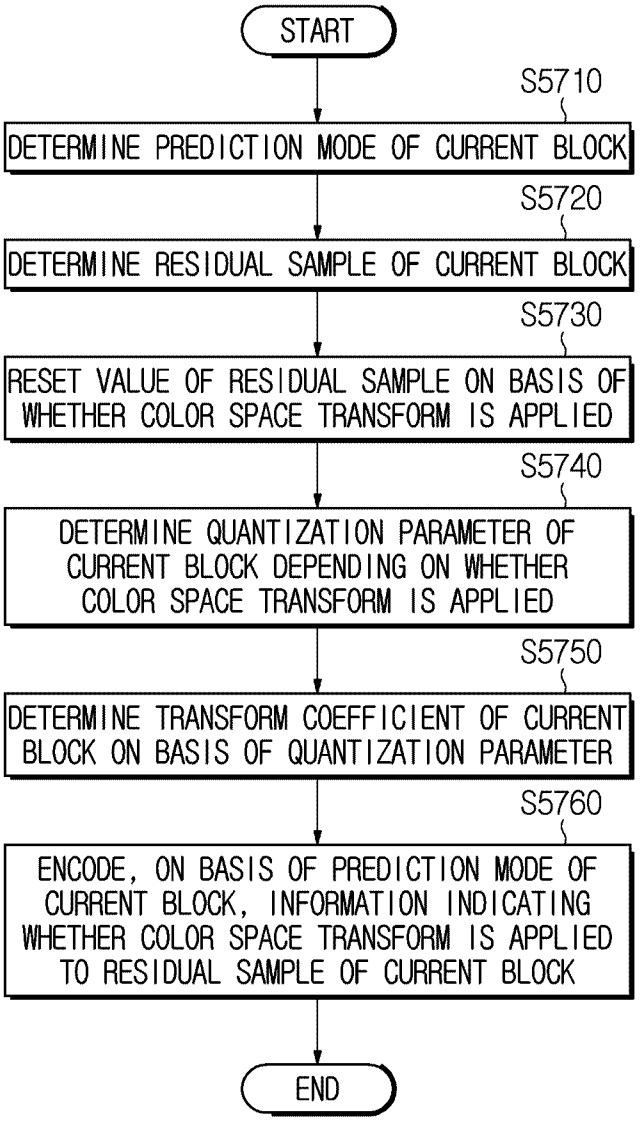
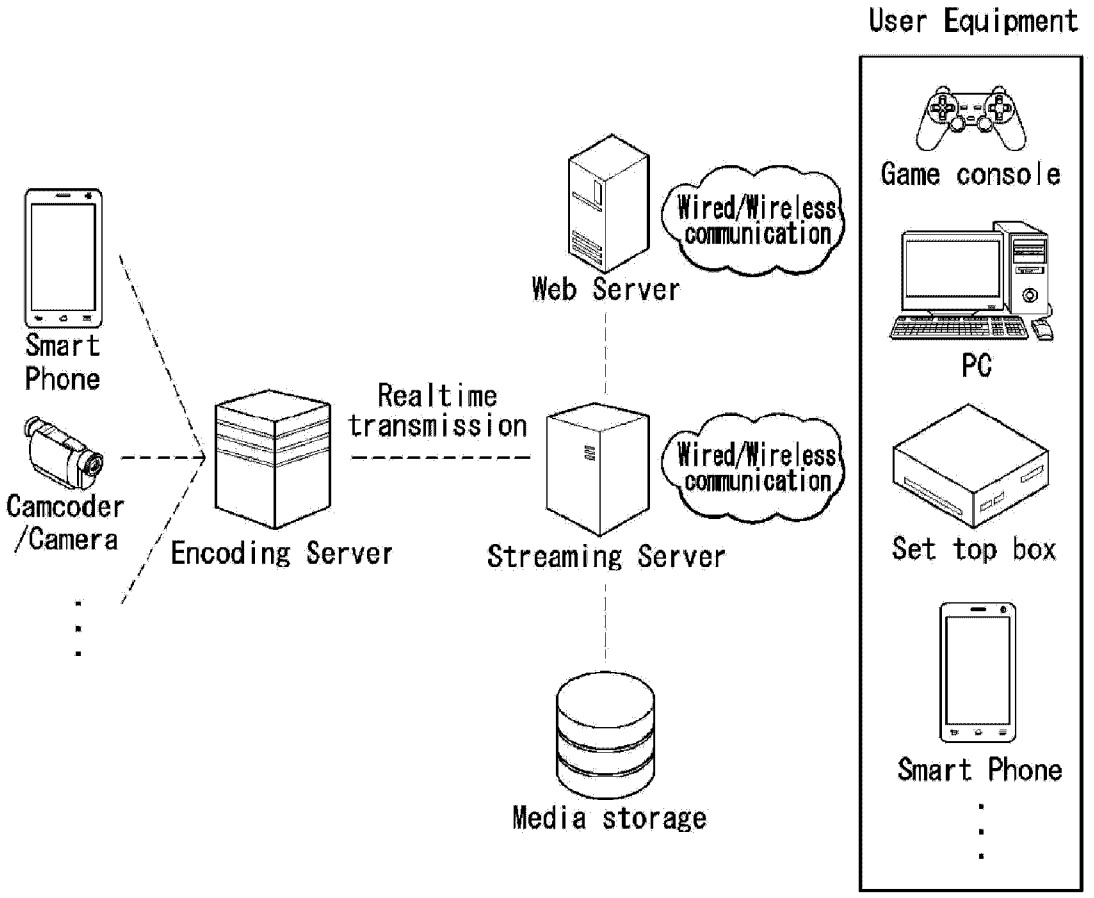

START
S5710
DETERMINE PREDICTION MODE OF CURRENT BLOCK
S5720
DETERMINE RESIDUAL SAMPLE OF CURRENT BLOCK
S5730
RESET VALUE OF RESIDUAL SAMPLE ON BASIS OF WHETHER COLOR SPACE TRANSFORM IS APPLIED
S5740
DETERMINE QUANTIZATION PARAMETER OF CURRENT BLOCK DEPENDING ON WHETHER COLOR SPACE TRANSFORM IS APPLIED
S5750
DETERMINE TRANSFORM COEFFICIENT OF CURRENT BLOCK ON BASIS OF QUANTIZATION PARAMETER
S5760
ENCODE, ON BASIS OF PREDICTION MODE OF CURRENT BLOCK, INFORMATION INDICATING WHETHER COLOR SPACE TRANSFORM IS APPLIED TO RESIDUAL SAMPLE OF CURRENT BLOCK
END

FIG. 58


User Equipment
Game console
Wired/Wireless communication
Web Server
Smart Phone
PC
Realtime transmission
Camcoder /Camera
Encoding Server
Streaming Server
Wired/Wireless communication
Set top box
Smart Phone
Media storage