

(19)



(11)

**EP 4 040 792 B9**

(12)

**CORRECTED EUROPEAN PATENT SPECIFICATION**

(15) Correction information:  
**Corrected version no 1 (W1 B1)**  
**Corrections, see**  
**Description Paragraph(s) 170, 259**

(51) International Patent Classification (IPC):  
**H04N 19/124** <sup>(2014.01)</sup> **H04N 19/157** <sup>(2014.01)</sup>  
**H04N 19/176** <sup>(2014.01)</sup> **H04N 19/61** <sup>(2014.01)</sup>  
**H04N 19/70** <sup>(2014.01)</sup>

(48) Corrigendum issued on:  
**16.07.2025 Bulletin 2025/29**

(52) Cooperative Patent Classification (CPC):  
**H04N 19/124; H04N 19/157; H04N 19/176;**  
**H04N 19/61; H04N 19/70**

(45) Date of publication and mention  
of the grant of the patent:  
**16.04.2025 Bulletin 2025/16**

(86) International application number:  
**PCT/KR2020/013495**

(21) Application number: **20872437.7**

(87) International publication number:  
**WO 2021/066618 (08.04.2021 Gazette 2021/14)**

(22) Date of filing: **05.10.2020**

(54) **IMAGE OR VIDEO CODING BASED ON SIGNALING OF TRANSFORM SKIP- AND PALETTE CODING-RELATED INFORMATION**

BILD- ODER VIDEOCODIERUNG BASIEREND AUF DEM SIGNALISIEREN VON TRANSFORMATIONS-SPRUNG- UND PALETTENCODIERUNGS-BEZOGENEN INFORMATIONEN  
CODAGE D'IMAGE OU DE VIDÉO BASÉ SUR LA SIGNALISATION D'INFORMATIONS LIÉES AU SAUT DE TRANSFORMÉE ET AU CODAGE DE PALETTE

(84) Designated Contracting States:  
**AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR**

(74) Representative: **Frenkel, Matthias Alexander Wuesthoff & Wuesthoff Patentanwälte und Rechtsanwalt PartG mbB Schweigerstraße 2 81541 München (DE)**

(30) Priority: **05.10.2019 US 201962911221 P**

(56) References cited:  
**KR-A- 20160 031 495 KR-A- 20170 140 145 KR-B1- 101 837 040**

(43) Date of publication of application:  
**10.08.2022 Bulletin 2022/32**

(60) Divisional application:  
**25170053.0 / 4 561 074**

- **BROSS B ET AL: "Versatile Video Coding (Draft 6)", no. JVET-O2001-vE; JVET-O2001, 31 July 2019 (2019-07-31), pages 1 - 455, XP030293944, Retrieved from the Internet <URL:https://jvet-experts.org/doc\_end\_user/documents/15\_Gothenburg/wg11/JVET-O2001-v14.zip JVET-O2001-vE.docx> [retrieved on 20190731]**
- **SARWER (ALIBABA-INC) M G ET AL: "Non-CE6: A clean-up of SPS maximum SBT size signaling", no. JVET-P0391, 24 September 2019 (2019-09-24), XP030217163, Retrieved from the Internet <URL:http://phenix.int-evry.fr/jvet/doc\_end\_user/documents/16\_Geneva/wg11/JVET-P0391-v1.zip JVET-P0391.docx> [retrieved on 20190924]**

(73) Proprietor: **LG Electronics Inc. SEOUL 07336 (KR)**

- (72) Inventors:
- **YOO, Sunmi Seoul 06772 (KR)**
  - **CHOI, Jungah Seoul 06772 (KR)**
  - **CHOI, Jangwon Seoul 06772 (KR)**

**EP 4 040 792 B9**

Note: Within nine months of the publication of the mention of the grant of the European patent in the European Patent Bulletin, any person may give notice to the European Patent Office of opposition to that patent, in accordance with the Implementing Regulations. Notice of opposition shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

- SARWER (ALIBABA-INC) M G ET AL: "CE8-related: Alignment of maximum transform-skip size with maximum transform block size", no. m50456, 25 September 2019 (2019-09-25), XP030206560, Retrieved from the Internet <URL:Moving the signaling of maximum transform skip size from PPS to SPShttp://phenix.int-evry.fr/mpeg/doc\_end\_user/documents/128\_Geneva/wg11/m50456-JVET-P0486-v1-JVET-P0486.zip JVET-P0486.docx> [retrieved on 20190925]
- KIM (WILUSGROUP) D ET AL: "AHG9: Cleanups on redundant signalling in HLS", no. JVET-Q0374 ; m51969, 31 December 2019 (2019-12-31), XP030223355, Retrieved from the Internet <URL:http://phenix.int-evry.fr/jvet/doc\_end\_user/documents/17\_Brussels/wg11/JVET-Q0374-v1.zip JVET-Q0374-v1.docx> [retrieved on 20191231]
- S-T HSIANG ET AL: "AHG9: High-level syntax related to transform skip mode", no. JVET-Q0183 ; m51772, 31 December 2019 (2019-12-31), XP030222817, Retrieved from the Internet <URL:http://phenix.int-evry.fr/jvet/doc\_end\_user/documents/17\_Brussels/wg11/JVET-Q0183-v1.zip JVET-Q0183-v1.docx> [retrieved on 20191231]
- ZHAO (LGE) J ET AL: "Non-CE8: Minimum QP for Palette Escape Coding", no. JVET-P0460 ; m50428, 25 September 2019 (2019-09-25), XP030217369, Retrieved from the Internet <URL:http://phenix.int-evry.fr/jvet/doc\_end\_user/documents/16\_Geneva/wg11/JVET-P0460-v1.zip JVET-P0460.docx> [retrieved on 20190925]
- Y.-H. CHAO (QUALCOMM), H. WANG, V. SEREGIN, M. KARCZEWICZ (QUALCOMM): "Non-CE8: Minimum QP for escape mode in palette", 16. JVET MEETING; 20191001 - 20191011; GENEVA; (THE JOINT VIDEO EXPLORATION TEAM OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16 ), 25 September 2019 (2019-09-25), XP030217412
- S. YOO (LGE), J. NAM, J. CHOI, J. LIM, S. KIM (LGE): "CE8-related : Unified condition between BDPCM and transform skip", 15. JVET MEETING; 20190703 - 20190712; GOTHENBURG; (THE JOINT VIDEO EXPLORATION TEAM OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16 ), 25 June 2019 (2019-06-25), XP030218844

**Description****BACKGROUND OF THE DISCLOSURE**5 Field of the disclosure

**[0001]** The present technology relates to video or image coding, and for example, to an image or video coding technology based on signaling of transform skip and palette coding related information.

10 Related Art

**[0002]** Recently, the demand for high resolution, high quality image/video such as 4K, 8K or more Ultra High Definition (UHD) image/video is increasing in various fields. As the image/video resolution or quality becomes higher, relatively more amount of information or bits are transmitted than for conventional image/video data. Therefore, if image/video data are transmitted via a medium such as an existing wired/wireless broadband line or stored in a legacy storage medium, costs for transmission and storage are readily increased.

**[0003]** Moreover, interests and demand are growing for virtual reality (VR) and artificial reality (AR) contents, and immersive media such as hologram; and broadcasting of images/videos exhibiting image/video characteristics different from those of an actual image/video, such as game images/videos, are also growing.

**[0004]** Therefore, a highly efficient image/video compression technique is required to effectively compress and transmit, store, or play high resolution, high quality images/videos showing various characteristics as described above.

**[0005]** Further, there is a need for a scheme for improving the overall image/video coding efficiency by efficiently classifying whether to code related information in accordance with dependency and non-dependency of information which is absolutely necessary or auxiliary used in performing transform skip and palette coding. BROSS B ET AL: "Versatile Video Coding (Draft 6)", 15. JVET MEETING; 20190703 - 20190712; GOTHENBURG; (THE JOINT VIDEO EXPLORATION TEAM OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16 ), no. JVET-02001-vE; JVET-02001 31 July 2019 (2019-07-31), pages 1-455, XP030293944, Retrieved from the Internet: URL:https://jvet-experts.org/doc\_end\_user/documents/15\_Gothenburg/wg11/JVET-02001-v14.zip JVET-02001-vE.docx [retrieved on 2019-07-31] discloses conditional signalling e.g. signalling syntax for enabling BDPCM based on transform skip enabled information at the SPS level. It does not disclose signalling minimum quantization parameter information based on a condition that the value of the transform skip enabled information is equal to 1 and a condition that the value of the palette coding enabled information is equal to 1.

**SUMMARY**

**[0006]** A technical subject of the present document is to provide a method and an apparatus for enhancing video/image coding efficiency.

**[0007]** Another technical subject of the present document is to provide a method and an apparatus for efficiently parsing/signaling transform skip and/or palette coding related information.

**[0008]** Still another technical subject of the present document is to provide a method and an apparatus for efficiently determining whether to perform coding in accordance with dependency and/or non-dependency of information being used during transform skip and/or palette coding.

**[0009]** Yet still another technical subject of the present document is to provide a method and an apparatus for defining a dependent condition for effectively parsing a syntax element having dependency with respect to a transform skip and/or palette coding related high-level syntax element, and determining whether to perform parsing based on the dependent condition.

**[0010]** According to an embodiment of the present document, transform skip enabled information and pallet enabled information may be signaled through a sequence parameter set (SPS), and whether to parse/signal minimum quantization parameter information related to the minimum allowed quantization parameter for a transform skip mode may be determined based on at least one of the transform skip enabled information and the palette enabled information. For example, the minimum quantization parameter information may be parsed/signaled through the SPS based on a condition that a value of the transform skip enabled information is 1 or a value of the palette coding enabled information is 1.

**[0011]** According to an embodiment of the present document, a video/image decoding method performed by a decoding apparatus is provided. The video/image decoding method may include a method disclosed in embodiments of the present document.

**[0012]** According to an embodiment of the present document, a decoding apparatus performing video/image decoding is provided. The decoding apparatus may perform a method disclosed in embodiments of the present document.

**[0013]** According to an embodiment of the present document, a video/image encoding method performed by an

encoding apparatus is provided. The video/image encoding method may include a method disclosed in embodiments of the present document.

[0014] According to an embodiment of the present document, an encoding apparatus performing video/image encoding is provided. The encoding apparatus may perform a method disclosed in embodiments of the present document.

[0015] According to an embodiment of the present document, a computer-readable digital storage medium storing encoded video/image information generated according to a video/image encoding method disclosed in at least one of embodiments of the present document is provided.

[0016] According to an embodiment of the present document, a computer-readable digital storage medium storing encoded information or encoded video/image information causing a decoding apparatus to perform a video/image decoding method disclosed in at least one of embodiments of the present document is provided.

[0017] The present document may have various effects. For example, according to an embodiment of the present document, the overall image/video compression efficiency can be enhanced. Further, according to an embodiment of the present document, the transform skip and/or palette coding related information can be efficiently parsed/signaled. Further, according to an embodiment of the present document, whether to perform coding can be effectively determined in accordance with the dependency and/or non-dependency of the information being used during the transform skip and/or palette coding. Further, according to an embodiment of the present document, efficient coding is possible by defining the dependent condition for effectively parsing the syntax element having dependency with respect to the transform skip and/or palette coding related high-level syntax element, and determining whether to perform parsing in accordance with the dependent condition. Further, according to an embodiment of the present document, bits being transmitted can be saved by determining whether to perform parsing in accordance with the dependent condition with respect to the transform skip and/or palette coding related high-level syntax element.

Said embodiments are meant to illustrate the invention and not to limit its scope: the scope of the invention is defined by the appended claims.

The invention is disclosed in figures 10, 12 in conjunction with their associated description. Sometimes the expressions "embodiment" and "invention" are used also with reference other teachings included in the description. In these cases they are to be interpreted as meaning "further example(s) not representing the invention", unless they are relative to subject-matter falling under the scope of the appended claims

### BRIEF DESCRIPTION OF THE DRAWINGS

[0018]

FIG. 1 schematically illustrates an example of a video/image coding system to which embodiments of the present document are applicable.

FIG. 2 is a diagram schematically illustrating the configuration of a video/image encoding apparatus to which embodiments of the present document are applicable.

FIG. 3 is a diagram schematically explaining the configuration of a video/image decoding apparatus to which embodiments of the present document are applicable.

FIG. 4 illustrates an example of a schematic video/image encoding method to which embodiments of the present document are applicable.

FIG. 5 illustrates an example of a schematic video/image decoding method to which embodiments of the present document are applicable.

FIG. 6 schematically illustrates an example of an entropy encoding method to which embodiments of the present document are applicable, and FIG. 7 schematically illustrates an entropy encoder in an encoding apparatus.

FIG. 8 schematically illustrates an example of an entropy decoding method to which embodiments of the present document are applicable, and FIG. 9 schematically illustrates an entropy decoder in a decoding apparatus.

FIGS. 10 and 11 schematically illustrate a video/image encoding method and an example of related components according to embodiment(s) of the present document.

FIGS. 12 and 13 schematically illustrate a video/image decoding method and an example of related components according to embodiment(s) of the present document.

FIG. 14 illustrates an example of a content streaming system to which embodiments disclosed in the present document are applicable.

### DESCRIPTION OF EXEMPLARY EMBODIMENTS

[0019] Terms commonly used in this specification are used to describe a specific embodiment and is not used to limit the technical spirit of the present document. An expression of the singular number includes plural expressions unless evidently expressed otherwise in the context. A term, such as "include" or "have" in this specification, should be

understood to indicate the existence of a characteristic, number, step, operation, element, part, or a combination of them described in the specification and not to exclude the existence or the possibility of the addition of one or more other characteristics, numbers, steps, operations, elements, parts or a combination of them.

**[0020]** Meanwhile, elements in the drawings described in the present document are independently illustrated for convenience of description related to different characteristic functions. This does not mean that each of the elements is implemented as separate hardware or separate software. For example, at least two of elements may be combined to form a single element, or a single element may be divided into a plurality of elements. An embodiment in which elements are combined and/or separated is also included in the scope of rights of the present document unless it deviates from the essence of the present document.

**[0021]** In the present document, the term "A or B" may mean "only A", "only B", or "both A and B". In other words, in the present document, the term "A or B" may be interpreted to indicate "A and/or B". For example, in the present document, the term "A, B or C" may mean "only A", "only B", "only C", or "any combination of A, B and C".

**[0022]** A slash "/" or a comma used in the present document may mean "and/or". For example, "A/B" may mean "A and/or B". Accordingly, "A/B" may mean "only A", "only B", or "both A and B". For example, "A, B, C" may mean "A, B or C".

**[0023]** In the present document, "at least one of A and B" may mean "only A", "only B", or "both A and B". Further, in the present document, the expression "at least one of A or B" or "at least one of A and/or B" may be interpreted the same as "at least one of A and B".

**[0024]** Further, in the present document, "at least one of A, B and C" may mean "only A", "only B", "only C", or "any combination of A, B and C". Further, "at least one of A, B or C" or "at least one of A, B and/or C" may mean "at least one of A, B and C".

**[0025]** Further, the parentheses used in the present document may mean "for example". Specifically, in the case that "prediction (intra prediction)" is expressed, it may be indicated that "intra prediction" is proposed as an example of "prediction". In other words, the term "prediction" in the present document is not limited to "intra prediction", and it may be indicated that "intra prediction" is proposed as an example of "prediction". Further, even in the case that "prediction (i.e., intra prediction)" is expressed, it may be indicated that "intra prediction" is proposed as an example of "prediction".

**[0026]** The present document relates to video/image coding. For example, the methods/embodiments disclosed in the present document are applied to a method disclosed in the versatile video coding (VVC) standard. Further, the methods/embodiments disclosed in the present document may be applied to a method disclosed in the essential video coding (EVC) standard, the AOMedia Video 1 (AV1) standard, the 2nd generation of audio video coding standard (AVS2), or the next generation video/image coding standard (ex. H.267 or H.268, etc.).

**[0027]** The present document presents various embodiments of video/image coding, and the embodiments may be performed in combination with each other unless otherwise mentioned.

**[0028]** In the present document, a video may mean a set of a series of images according to the passage of time. A picture generally means a unit representing one image in a specific time period, and a slice/tile is a unit constituting a part of the picture in coding. The slice/tile may include one or more coding tree units (CTUs). One picture may consist of one or more slices/tiles. A tile is a rectangular region of CTUs within a particular tile column and a particular tile row in a picture. The tile column is a rectangular region of CTUs having a height equal to the height of the picture and a width specified by syntax elements in the picture parameter set. The tile row is a rectangular region of CTUs having a width specified by syntax elements in the picture parameter set and a height equal to the height of the picture. A tile scan is a specific sequential ordering of CTUs partitioning a picture in which the CTUs are ordered consecutively in CTU raster scan in a tile whereas tiles in a picture are ordered consecutively in a raster scan of the tiles of the picture. A slice includes an integer number of complete tiles or an integer number of consecutive complete CTU rows within a tile of a picture that may be exclusively contained in a single NAL unit.

**[0029]** Meanwhile, one picture may be divided into two or more subpictures. The subpicture may be a rectangular region of one or more slices within the picture.

**[0030]** A pixel or a pel may mean a smallest unit constituting one picture (or image). Also, '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. Alternatively, a sample may mean a pixel value in the spatial domain, or may mean a transform coefficient in the frequency domain when the pixel value is transformed into the frequency domain.

**[0031]** 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 (ex. cb, cr) blocks. The unit may be used interchangeably with terms such as block or area in some cases. In a general case, an  $M \times N$  block may include samples (or sample arrays) or a set (or array) of transform coefficients of M columns and N rows.

**[0032]** Also, in the present document, at least one of quantization/dequantization and/or transform/inverse transform may be omitted. When the quantization/dequantization is omitted, the quantized transform coefficient may be referred to as a transform coefficient. When the transform/inverse transform is omitted, transform coefficients may be called coefficients or residual coefficients, or may still be called transform coefficients for the sake of uniformity of expression.

**[0033]** In the present document, a quantized transform coefficient and a transform coefficient may be referred to as a transform coefficient and a scaled transform coefficient, respectively. In this case, the residual information may include information about the transform coefficient(s), and the information about the transform coefficient(s) may be signaled through a residual coding syntax. Transform coefficients may be derived based on residual information (or information about transform coefficient(s)), and scaled transform coefficients may be derived through inverse transform (scaling) on the transform coefficients. Residual samples may be derived based on an inverse transform (transform) for the scaled transform coefficients. This may be applied/expressed in other parts of the present document as well.

**[0034]** In the present document, technical features individually explained in one drawing may be individually implemented, or may be simultaneously implemented.

**[0035]** Hereinafter, preferred embodiments of the present document are described more specifically with reference to the accompanying drawings. Hereinafter, in the drawings, the same reference numeral is used in the same element, and a redundant description of the same element may be omitted.

**[0036]** FIG. 1 illustrates an example of a video/image coding system to which the embodiments of the present document may be applied.

**[0037]** Referring to FIG. 1, a video/image coding system may include a source device and a reception device. The source device may transmit encoded video/image information or data to the reception device through a digital storage medium or network in the form of a file or streaming.

**[0038]** The source device may include a video source, an encoding apparatus, and a transmitter. The receiving device may include a receiver, a decoding apparatus, and a renderer. The encoding apparatus may be called a video/image encoding apparatus, and the decoding apparatus may be called a video/image decoding apparatus. The transmitter may be included in the encoding apparatus. The receiver may be included in the decoding apparatus. The renderer may include a display, and the display may be configured as a separate device or an external component.

**[0039]** The video source may acquire video/image through a process of capturing, synthesizing, or generating the video/image. The video source 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.

**[0040]** The encoding apparatus may encode input video/image. The encoding apparatus may perform a series of procedures such as prediction, transform, and quantization for compaction and coding efficiency. The encoded data (encoded video/image information) may be output in the form of a bitstream.

**[0041]** The transmitter may transmit the encoded image/image information or data output in the form of a bitstream to the receiver of the receiving device 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 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 may receive/extract the bitstream and transmit the received bitstream to the decoding apparatus.

**[0042]** The decoding apparatus 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.

**[0043]** The renderer may render the decoded video/image. The rendered video/image may be displayed through the display.

**[0044]** FIG. 2 is a diagram schematically illustrating a configuration of a video/image encoding apparatus to which the embodiments of the present document may be applied. Hereinafter, what is referred to as the encoding apparatus may include an image encoding apparatus and/or a video encoding apparatus.

**[0045]** Referring to FIG. 2, the encoding apparatus 200 may include and be configured with an image partitioner 210, a predictor 220, a residual processor 230, an entropy encoder 240, an adder 250, a filter 260, and a memory 270. The predictor 220 may include an inter predictor 221 and an intra predictor 222. The residual processor 230 may include a transformer 232, a quantizer 233, a dequantizer 234, and an inverse transformer 235. The residual processor 230 may further include a subtractor 231. The adder 250 may be called a reconstructor or reconstructed block generator. The image partitioner 210, the predictor 220, the residual processor 230, the entropy encoder 240, the adder 250, and the filter 260, which have been described above, may be configured by one or more hardware components (e.g., encoder chipsets or processors) according to an embodiment. In addition, the memory 270 may include a decoded picture buffer (DPB), and may also be configured by a digital storage medium. The hardware component may further include the memory 270 as an internal/external component.

**[0046]** The image partitioner 210 may split an input image (or, picture, frame) input to the encoding apparatus 200 into one or more processing units. As an example, the processing unit may be called a coding unit (CU). In this case, the coding unit may be recursively split according to a Quad-tree binary-tree ternary-tree (QTBT) structure from a coding tree unit

(CTU) or the largest coding unit (LCU). For example, one coding unit may be split into a plurality of coding units of a deeper depth based on a quad-tree structure, a binary-tree structure, and/or a ternary-tree structure. In this case, for example, the quad-tree structure is first applied and the binary-tree structure and/or the ternary-tree structure may be later applied. Alternatively, the binary-tree structure may also be first applied. A coding procedure according to the present document may be performed based on a final coding unit which is not split any more. In this case, based on coding efficiency according to image characteristics or the like, the maximum coding unit may be directly used as the final coding unit, or as necessary, the coding unit may be recursively split into coding units of a deeper depth, such that a coding unit having an optimal size may be used as the final coding unit. Here, the coding procedure may include a procedure such as prediction, transform, and reconstruction to be described later. As another example, the processing unit may further include a prediction unit (PU) or a transform unit (TU). In this case, each of the prediction unit and the transform unit may be split or partitioned from the aforementioned final coding unit. The prediction unit may be a unit of sample prediction, and the transform unit may be a unit for inducing a transform coefficient and/or a unit for inducing a residual signal from the transform coefficient.

**[0047]** The unit may be interchangeably used with the term such as a block or an area in some cases. Generally, an MxN block may represent samples composed of M columns and N rows or a group of transform coefficients. The sample may generally represent a pixel or a value of the pixel, and may also represent only the pixel/pixel value of a luma component, and also represent only the pixel/pixel value of a chroma component. The sample may be used as the term corresponding to a pixel or a pel configuring one picture (or image).

**[0048]** The encoding apparatus 200 may generate a residual signal (residual block, residual sample array) by subtracting a predicted signal (predicted block, prediction sample array) output from the inter predictor 221 or the intra predictor 222 from the input image signal (original block, original sample array), and the generated residual signal is transmitted to the transformer 232. In this case, as illustrated, the unit for subtracting the predicted signal (predicted block, prediction sample array) from the input image signal (original block, original sample array) within an encoder 200 may be called the subtractor 231. The predictor may perform prediction for a block to be processed (hereinafter, referred to as a current block), and generate a predicted block including prediction samples of the current block. The predictor may determine whether intra prediction is applied or inter prediction is applied in units of the current block or the CU. The predictor may generate various information about prediction, such as prediction mode information, to transfer the generated information to the entropy encoder 240 as described later in the description of each prediction mode. The information about prediction may be encoded by the entropy encoder 240 to be output in a form of the bitstream.

**[0049]** The intra predictor 222 may predict a current block with reference to samples within a current picture. The referenced samples may be located neighboring to the current block, or may also be located away from the current block according to the prediction mode. The prediction modes in the intra prediction 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 or a planar mode. The directional mode may include, for example, 33 directional prediction modes or 65 directional prediction modes according to the fine degree of the prediction direction. However, this is illustrative and the directional prediction modes which are more or less than the above number may be used according to the setting. The intra predictor 222 may also determine the prediction mode applied to the current block using the prediction mode applied to the neighboring block.

**[0050]** The inter predictor 221 may induce a predicted block of the current block based on a reference block (reference sample array) specified by a motion vector on a reference picture. At this time, in order to decrease the amount of motion information transmitted in the inter prediction mode, the motion information may be predicted in units of a block, a sub-block, or a sample based on the correlation of the 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, or the like) information. In the case of the inter prediction, the neighboring block may include a spatial neighboring block existing within the current picture and a temporal neighboring block existing in the reference picture. The reference picture including the reference block and the reference picture including the temporal neighboring block may also be the same as each other, and may also be different from each other. The temporal neighboring block may be called the name such as a collocated reference block, a collocated CU (colCU), or the like, and the reference picture including the temporal neighboring block may also be called a collocated picture (colPic). For example, the inter predictor 221 may configure a motion information candidate list based on the neighboring blocks, and generate information indicating what candidate is used to derive the motion vector and/or the reference picture index of the current block. The inter prediction may be performed based on various prediction modes, and for example, in the case of a skip mode and a merge mode, the inter predictor 221 may use the motion information of the neighboring block as the motion information of the current block. In the case of the skip mode, the residual signal may not be transmitted unlike the merge mode. A motion vector prediction (MVP) mode may indicate the motion vector of the current block by using the motion vector of the neighboring block as a motion vector predictor, and signaling a motion vector difference.

**[0051]** The predictor 200 may generate a predicted signal based on various prediction methods to be described later. For example, the predictor may not only apply the intra prediction or the inter prediction for predicting one block, but also

simultaneously apply the intra prediction and the inter prediction. This may be called a combined inter and intra prediction (CIIP). Further, the predictor may be based on an intra block copy (IBC) prediction mode, or a palette mode in order to perform prediction on a block. The IBC prediction mode or palette mode may be used for content image/video coding of a game or the like, such as screen content coding (SCC). The IBC basically performs prediction in a current picture, but it may be performed similarly to inter prediction in that it derives a reference block in a current picture. That is, the IBC may use at least one of inter prediction techniques described in the present document. The palette mode may be regarded as an example of intra coding or intra prediction. When the palette mode is applied, a sample value in a picture may be signaled based on information on a palette index and a palette table.

**[0052]** The predicted signal generated through the predictor (including the inter predictor 221 and/or the intra predictor 222) may be used to generate a reconstructed signal or used to generate a residual signal. The transformer 232 may generate transform coefficients by applying the 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 nonlinear transform (CNT). Here, when the relationship information between pixels is illustrated as a graph, the GBT means the transform obtained from the graph. The CNT means the transform which is acquired based on a predicted signal generated by using all previously reconstructed pixels. In addition, the transform process may also be applied to a pixel block having the same size of the square, and may also be applied to the block having a variable size rather than the square.

**[0053]** The quantizer 233 may quantize the transform coefficients to transmit the quantized transform coefficients to the entropy encoder 240, and the entropy encoder 240 may encode the quantized signal (information about the quantized transform coefficients) to the encoded quantized signal to the bitstream. The information about the quantized transform coefficients may be called residual information. The quantizer 233 may rearrange the quantized transform coefficients having a block form in a one-dimensional vector form based on a coefficient scan order, and also generate the information about the quantized transform coefficients based on the quantized transform coefficients of the one dimensional vector form. The entropy encoder 240 may perform various encoding methods, for example, such as an exponential Golomb coding, a context-adaptive variable length coding (CAVLC), and a context-adaptive binary arithmetic coding (CABAC). The entropy encoder 240 may also encode information (e.g., values of syntax elements and the like) necessary for reconstructing video/image other than the quantized transform coefficients together or separately. The encoded information (e.g., encoded video/image information) may be transmitted or stored in units of network abstraction layer (NAL) unit in a form of the bitstream. The video/image information may further include information about 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/transmitted information and/or syntax elements to be described later in this document may be encoded through the aforementioned encoding procedure and thus included in the bitstream. The bitstream may be transmitted through a network, or stored in a digital storage medium. Here, the network may include a broadcasting network and/or a communication network, or the like, and the digital storage medium may include various storage media such as USB, SD, CD, DVD, Blue-ray, HDD, and SSD. A transmitter (not illustrated) for transmitting the signal output from the entropy encoder 240 and/or a storage (not illustrated) for storing the signal may be configured as the internal/external elements of the encoding apparatus 200, or the transmitter may also be included in the entropy encoder 240.

**[0054]** The quantized transform coefficients output from the quantizer 233 may be used to generate a predicted signal. For example, the dequantizer 234 and the inverse transformer 235 apply dequantization and inverse transform to the quantized transform coefficients, such that the residual signal (residual block or residual samples) may be reconstructed. The adder 250 adds the reconstructed residual signal to the predicted signal output from the inter predictor 221 or the intra predictor 222, such that the reconstructed signal (reconstructed picture, reconstructed block, reconstructed sample array) may be generated. As in the case where the skip mode is applied, if there is no residual for the block to be processed, the predicted block may be used as the reconstructed block. The adder 250 may be called a reconstructor or a reconstructed block generator. The generated reconstructed signal may be used for the intra prediction of the next block to be processed within the current picture, and as described later, also used for the inter prediction of the next picture through filtering.

**[0055]** Meanwhile, a luma mapping with chroma scaling (LMCS) may also be applied in a picture encoding and/or reconstruction process.

**[0056]** The filter 260 may apply filtering to the reconstructed signal, thereby improving subjective/objective image qualities. For example, the filter 260 may apply various filtering methods to the reconstructed picture to generate a modified reconstructed picture, and store the modified reconstructed picture in the memory 270, specifically, the DPB of the memory 270. Various filtering methods may include, for example, a deblocking filtering, a sample adaptive offset, an adaptive loop filter, a bilateral filter, and the like. The filter 260 may generate various filtering-related information to transfer the generated information to the entropy encoder 240, as described later in the description of each filtering method. The filtering-related information may be encoded by the entropy encoder 240 to be output in a form of the bitstream.

**[0057]** The modified reconstructed picture transmitted to the memory 270 may be used as the reference picture in the inter predictor 221. If the inter prediction is applied by the inter predictor, the encoding apparatus may avoid the prediction

mismatch between the encoding apparatus 200 and the decoding apparatus, and also improve coding efficiency.

**[0058]** The DPB of the memory 270 may store the modified reconstructed picture to be used as the reference picture in the inter predictor 221. The memory 270 may store motion information of the block in which the motion information within the current picture is derived (or encoded) and/or motion information of the blocks within the previously reconstructed picture. The stored motion information may be transferred to the inter predictor 221 to be utilized as motion information of the spatial neighboring block or motion information of the temporal neighboring block. The memory 270 may store the reconstructed samples of the reconstructed blocks within the current picture, and transfer the reconstructed samples to the intra predictor 222.

**[0059]** FIG. 3 is a diagram for schematically explaining a configuration of a video/image decoding apparatus to which the present document is applicable. Hereinafter, what is referred to as the decoding apparatus may include an image decoding apparatus and/or a video decoding apparatus.

**[0060]** Referring to FIG. 3, the decoding apparatus 300 may include and configured with an entropy decoder 310, a residual processor 320, a predictor 330, an adder 340, a filter 350, and a memory 360. The predictor 330 may include an inter predictor 331 and an intra predictor 332. The residual processor 320 may include a dequantizer 321 and an inverse transformer 322. The entropy decoder 310, the residual processor 320, the predictor 330, the adder 340, and the filter 350, which have been described above, may be configured by one or more hardware components (e.g., decoder chipsets or processors) according to an embodiment. Further, the memory 360 may include a decoded picture buffer (DPB), and may be configured by a digital storage medium. The hardware component may further include the memory 360 as an internal/external component.

**[0061]** When the bitstream including the video/image information is input, the decoding apparatus 300 may reconstruct the image in response to a process in which the video/image information is processed in the encoding apparatus illustrated in FIG. 2. For example, the decoding apparatus 300 may derive the units/blocks based on block split-related information acquired from the bitstream. The decoding apparatus 300 may perform decoding using the processing unit applied to the encoding apparatus. Therefore, the processing unit for the decoding may be, for example, a coding unit, and the coding unit may be split according to the quad-tree structure, the binary-tree structure, and/or the ternary-tree structure from the coding tree unit or the maximum coding unit. One or more transform units may be derived from the coding unit. In addition, the reconstructed image signal decoded and output through the decoding apparatus 300 may be reproduced through a reproducing apparatus.

**[0062]** The decoding apparatus 300 may receive the signal output from the encoding apparatus illustrated in FIG. 2 in a form of the bitstream, and the received signal may be decoded through the entropy decoder 310. For example, the entropy decoder 310 may derive information (e.g., video/image information) necessary for the image reconstruction (or picture reconstruction) by parsing the bitstream. The video/image information may further include information about various parameter sets such as an adaptation parameter set (APS), a picture parameter set (PPS), a sequence parameter set (SPS), and a video parameter set (VPS). In addition, the video/image information may further include general constraint information. The decoding apparatus may decode the picture further based on the information about the parameter set and/or the general constraint information. The signaled/received information and/or syntax elements to be described later in this document may be decoded through the decoding procedure and acquired from the bitstream. For example, the entropy decoder 310 may decode information within the bitstream based on a coding method such as an exponential Golomb coding, a CAVLC, or a CABAC, and output a value of the syntax element necessary for the image reconstruction, and the quantized values of the residual-related transform coefficient. More specifically, the CABAC entropy decoding method may receive a bin corresponding to each syntax element from the bitstream, determine a context model using syntax element information to be decoded and decoding information of the neighboring block and the block to be decoded or information of the symbol/bin decoded in the previous stage, and generate a symbol corresponding to a value of each syntax element by predicting the probability of generation of the bin according to the determined context model to perform the arithmetic decoding of the bin. At this time, the CABAC entropy decoding method may determine the context model and then update the context model using the information of the decoded symbol/bin for a context model of a next symbol/bin. The information about prediction among the information decoded by the entropy decoder 310 may be provided to the predictor (the inter predictor 332 and the intra predictor 331), and a residual value at which the entropy decoding is performed by the entropy decoder 310, that is, the quantized transform coefficients and the related parameter information may be input to the residual processor 320. The residual processor 320 may derive a residual signal (residual block, residual samples, residual sample array). In addition, the information about filtering among the information decoded by the entropy decoder 310 may be provided to the filter 350. Meanwhile, a receiver (not illustrated) for receiving the signal output from the encoding apparatus may be further configured as the internal/external element of the decoding apparatus 300, or the receiver may also be a component of the entropy decoder 310. Meanwhile, the decoding apparatus according to this document may be called a video/image/picture decoding apparatus, and the decoding apparatus may also 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 310, and the sample decoder may include at least one of the dequantizer 321, the inverse transformer 322, the adder 340, the filter 350, the memory 360, the inter

predictor 332, and the intra predictor 331.

**[0063]** The dequantizer 321 may dequantize the quantized transform coefficients to output the transform coefficients. The dequantizer 321 may rearrange the quantized transform coefficients in a two-dimensional block form. In this case, the rearrangement may be performed based on a coefficient scan order performed by the encoding apparatus. The dequantizer 321 may perform dequantization for the quantized transform coefficients using a quantization parameter (e.g., quantization step size information), and acquire the transform coefficients.

**[0064]** The inverse transformer 322 inversely transforms the transform coefficients to acquire the residual signal (residual block, residual sample array).

**[0065]** The predictor 330 may perform the prediction of the current block, and generate a predicted block including the prediction samples of the current block. The predictor may determine whether the intra prediction is applied or the inter prediction is applied to the current block based on the information about prediction output from the entropy decoder 310, and determine a specific intra/inter prediction mode.

**[0066]** The predictor may generate the predicted signal based on various prediction methods to be described later. For example, the predictor may not only apply the intra prediction or the inter prediction for the prediction of one block, but also apply the intra prediction and the inter prediction at the same time. This may be called a combined inter and intra prediction (CIIP). Further, the predictor may be based on an intra block copy (IBC) prediction mode, or a palette mode in order to perform prediction on a block. The IBC prediction mode or palette mode may be used for content image/video coding of a game or the like, such as screen content coding (SCC). The IBC basically performs prediction in a current picture, but it may be performed similarly to inter prediction in that it derives a reference block in a current picture. That is, the IBC may use at least one of inter prediction techniques described in the present document. The palette mode may be regarded as an example of intra coding or intra prediction. When the palette mode is applied, information on a palette table and a palette index may be included in the video/image information and signaled.

**[0067]** The intra predictor 331 may predict the current block with reference to the samples within the current picture. The referenced samples may be located neighboring to the current block according to the prediction mode, or may also be located away from the current block. The prediction modes in the intra prediction may include a plurality of non-directional modes and a plurality of directional modes. The intra predictor 331 may also determine the prediction mode applied to the current block using the prediction mode applied to the neighboring block.

**[0068]** The inter predictor 332 may induce the predicted block of the current block based on the reference block (reference sample array) specified by the motion vector on the reference picture. At this time, in order to decrease the amount of the motion information transmitted in the inter prediction mode, the motion information may be predicted in units of a block, a sub-block, or a sample based on the correlation of the 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, or the like) information. In the case of the inter prediction, the neighboring block may include a spatial neighboring block existing within the current picture and a temporal neighboring block existing in the reference picture. For example, the inter predictor 332 may configure a motion information candidate list based on the neighboring blocks, and derive the motion vector and/or the reference picture index of the current block based on received candidate selection information. The inter prediction may be performed based on various prediction modes, and the information about the prediction may include information indicating the mode of the inter prediction of the current block.

**[0069]** The adder 340 may add the acquired residual signal to the predicted signal (predicted block, prediction sample array) output from the predictor (including the inter predictor 332 and/or the intra predictor 331) to generate the reconstructed signal (reconstructed picture, reconstructed block, reconstructed sample array). As in the case where the skip mode is applied, if there is no residual for the block to be processed, the predicted block may be used as the reconstructed block.

**[0070]** The adder 340 may be called a reconstructor or a reconstructed block generator. The generated reconstructed signal may be used for the intra prediction of a next block to be processed within the current picture, and as described later, may also be output through filtering or may also be used for the inter prediction of a next picture.

**[0071]** Meanwhile, a luma mapping with chroma scaling (LMCS) may also be applied in the picture decoding process.

**[0072]** The filter 350 may apply filtering to the reconstructed signal, thereby improving the subjective/objective image qualities. For example, the filter 350 may apply various filtering methods to the reconstructed picture to generate a modified reconstructed picture, and transmit the modified reconstructed picture to the memory 360, specifically, the DPB of the memory 360. Various filtering methods may include, for example, a deblocking filtering, a sample adaptive offset, an adaptive loop filter, a bidirectional filter, and the like.

**[0073]** The (modified) reconstructed picture stored in the DPB of the memory 360 may be used as the reference picture in the inter predictor 332. The memory 360 may store motion information of the block in which the motion information within the current picture is derived (decoded) and/or motion information of the blocks within the previously reconstructed picture. The stored motion information may be transferred to the inter predictor 260 to be utilized as motion information of the spatial neighboring block or motion information of the temporal neighboring block. The memory 360 may store the

reconstructed samples of the reconstructed blocks within the current picture, and transfer the stored reconstructed samples to the intra predictor 331.

**[0074]** In the present document, the exemplary embodiments described in the filter 260, the inter predictor 221, and the intra predictor 222 of the encoding apparatus 200 may be applied equally to or to correspond to the filter 350, the inter predictor 332, and the intra predictor 331 of the decoding apparatus 300, respectively.

**[0075]** Meanwhile, as described above, in performing video coding, prediction is performed to improve compression efficiency. Through this, a predicted block including prediction samples for a current block as a block to be coded (i.e., a coding target block) may be generated. Here, the predicted block includes prediction samples in a spatial domain (or pixel domain). The predicted block is derived in the same manner in an encoding apparatus and a decoding apparatus, and the encoding apparatus may signal information (residual information) on residual between the original block and the predicted block, rather than an original sample value of an original block, to the decoding apparatus, thereby increasing image coding efficiency. The decoding apparatus may derive a residual block including residual samples based on the residual information, add the residual block and the predicted block to generate reconstructed blocks including reconstructed samples, and generate a reconstructed picture including the reconstructed blocks.

**[0076]** The residual information may be generated through a transform and quantization procedure. For example, the encoding apparatus may derive a residual block between the original block and the predicted block, perform a transform procedure on residual samples (residual sample array) included in the residual block to derive transform coefficients, perform a quantization procedure on the transform coefficients to derive quantized transform coefficients, and signal related residual information to the decoding apparatus (through a bit stream). Here, the residual information may include value information of the quantized transform coefficients, location information, a transform technique, a transform kernel, a quantization parameter, and the like. The decoding apparatus may perform dequantization/inverse transform procedure based on the residual information and derive residual samples (or residual blocks). The decoding apparatus may generate a reconstructed picture based on the predicted block and the residual block. Also, for reference for inter prediction of a picture afterward, the encoding apparatus may also dequantize/inverse-transform the quantized transform coefficients to derive a residual block and generate a reconstructed picture based thereon.

**[0077]** The following drawing has been prepared to explain a detailed example of the present document. Since the name of a detailed device or a detailed term or name (e.g., name of syntax) described in the drawing is exemplarily presented, the technical features of the present document are not limited to the detailed name used in the drawing.

**[0078]** FIG. 4 illustrates an example of a schematic video/image encoding method to which embodiments of the present document are applicable.

**[0079]** The method disclosed in FIG. 4 may be performed by the encoding apparatus 200 of FIG. 2 as described above. Specifically, S400 may be performed by the inter predictor 221 or the intra predictor 222 of the encoding apparatus 200, and S410, S420, S430, and S440 may be performed by the subtractor 231, the transformer 232, the quantizer 233, and the entropy encoder 240 of the encoding apparatus 200, respectively.

**[0080]** Referring to FIG. 4, the encoding apparatus may derive prediction samples through prediction for the current block (S400). The encoding apparatus may determine whether to perform inter prediction or intra prediction with respect to the current block, and may determine a detailed inter prediction mode or a detailed intra prediction mode based on an RD cost. In accordance with the determined mode, the encoding apparatus may derive the prediction samples for the current block.

**[0081]** The encoding apparatus may derive residual samples through comparison of the prediction samples with the original samples for the current block (S410).

**[0082]** The encoding apparatus may derive transform coefficients through a transform process for residual samples (S420), and may derive quantized transform coefficients by quantizing the derived transform coefficients (S430).

**[0083]** The quantization may be performed based on a quantization parameter. The transform process and/or the quantization process may be omitted. In case that the transform process is omitted, (quantized) (residual) coefficients for the residual samples may be coded in accordance with a residual coding technique to be described later. For unity of terms, even the (quantized) (residual) coefficient may be called a (quantized) transform coefficient.

**[0084]** The encoding apparatus may encode image information including prediction information and residual information, and may output the encoded image information in the form of a bitstream (S440). The prediction information may be information related to the prediction process, and may include information (e.g., in case that inter prediction is applied) about prediction mode information and motion information. The residual information may include information on the quantized transform coefficients. The residual information may be entropy-coded. Alternatively, the residual information may include information on the (quantized) (residual) coefficients.

**[0085]** The output bitstream may be transferred to a decoding apparatus through a storage medium or a network.

**[0086]** FIG. 5 illustrates an example of a schematic video/image decoding procedure to which embodiments of the present document are applicable.

**[0087]** The method disclosed in FIG. 5 may be performed by the decoding apparatus 300 of FIG. 3 as described above. Specifically, S500 may be performed by the inter predictor 332 or the intra predictor 331 of the decoding apparatus 300. In

S500, a process of deriving values of related syntax elements by decoding prediction information included in the bitstream may be performed by the entropy decoder 310 of the decoding apparatus 300. S510, S520, S530, and S540 may be performed by the entropy decoder 310, the dequantizer 321, the inverse transformer 322, and the adder 340 of the decoding apparatus 300, respectively.

5 **[0088]** Referring to FIG. 5, the decoding apparatus may perform an operation corresponding to the operation performed by the encoding apparatus. The decoding apparatus may perform inter prediction or intra prediction with respect to the current block based on the received prediction information, and may derive prediction samples (S500).

**[0089]** The decoding apparatus may derive quantized transform coefficients for the current block based on the received residual information (S510). The decoding apparatus may derive the quantized transform coefficients from the residual information through entropy decoding.

10 **[0090]** The decoding apparatus may derive the transform coefficients by dequantizing the quantized transform coefficients (S520). The dequantization may be performed based on the quantization parameter.

**[0091]** The decoding apparatus may derive residual samples through a dequantization process for the transform coefficients (S530).

15 **[0092]** The inverse transform process and/or the dequantization process may be omitted. In case that the inverse transform process is omitted, (quantized) (residual) coefficients may be derived from the residual information, and the residual samples may be derived based on the (quantized) (residual) coefficients.

**[0093]** The decoding apparatus may generate reconstructed samples for the current block based on the prediction samples and the residual samples, and based on this, may generate a reconstructed picture (S540). Thereafter, an in-loop filtering process may be further applied to the reconstructed picture as described above.

20 **[0094]** Meanwhile, as described above, the encoding apparatus may perform entropy encoding based on various encoding methods, for example, such as exponential Golomb, context-adaptive variable length coding (CAVLC), and context-adaptive binary arithmetic coding (CABAC). Further, the decoding apparatus may perform entropy decoding based on the coding method, such as the exponential Golomb coding, CAVLC, or CABAC. Hereinafter, an entropy encoding/decoding process will be described.

25 **[0095]** FIG. 6 schematically illustrates an example of an entropy encoding method to which embodiments of the present document are applicable, and FIG. 7 schematically illustrates an entropy encoder in an encoding apparatus. The entropy encoder in the encoding apparatus of FIG. 7 may be applied equally or correspondingly even to the entropy encoder 240 of the encoding apparatus 200 of FIG. 2 as described above.

30 **[0096]** Referring to FIGS. 6 and 7, the encoding apparatus (entropy encoder) may perform an entropy coding process for 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, inter prediction mode information, and the like), residual information, and in-loop filtering related information, and may also include various syntax elements thereof. The entropy coding may be performed in the unit of a syntax element. S600 to S610 may be performed by the entropy encoder 240 of the encoding apparatus 200 of FIG. 2 as described above.

35 **[0097]** The encoding apparatus may perform binarization for a target syntax element (S600). Here, the binarization may be based on various binarization methods, such as truncated rice binarization process and fixed-length binarization process, and the binarization method for the target syntax element may be predefined. The binarization process may be performed by the binarizer 242 in the entropy encoder 240.

40 **[0098]** The encoding apparatus may perform entropy encoding for the target syntax element (S610). The encoding apparatus may perform normal coding based (context-based) or bypass coding based encoding of a bin string of the target syntax element based on an entropy coding technique, such as context-adaptive arithmetic coding (CABAC) or context-adaptive variable length coding (CAVLC), and its output may be included in the bitstream. The entropy encoding process may be performed by the entropy encoding processor 243 in the entropy encoder 240. The bitstream may be transferred to the decoding apparatus through a (digital) storage medium or a network as described above.

45 **[0099]** FIG. 8 schematically illustrates an example of an entropy decoding method to which embodiments of the present document are applicable, and FIG. 9 schematically illustrates an entropy decoder in a decoding apparatus. The entropy decoder in the decoding apparatus of FIG. 9 may be applied equally or correspondingly even to the entropy decoder 310 of the decoding apparatus 300 of FIG. 3 as described above.

50 **[0100]** Referring to FIGS. 8 and 9, 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, inter prediction mode information, and the like), residual information, and in-loop filtering related information, and may also include various syntax elements thereof. The entropy coding may be performed in the unit of a syntax element. S800 to S810 may be performed by the entropy decoder 310 of the decoding apparatus 300 of FIG. 3 as described above.

55 **[0101]** The decoding apparatus may perform binarization for a target syntax element (S800). Here, the binarization may be based on various binarization methods, such as truncated rice binarization process and fixed-length binarization process, and the binarization method for the target syntax element may be predefined. The decoding apparatus may

derive enabled bin strings (bin string candidates) for enabled values of the target syntax element through the binarization process. The binarization process may be performed by the binarizer 312 in the entropy decoder 310.

[0102] The decoding apparatus may perform entropy decoding for the target syntax element (S810). The decoding apparatus compares the derived bin string with enabled bin strings for the corresponding syntax element while sequentially decoding and parsing respective bins for the target syntax element from input bit(s) in the bitstream. If the derived bin string is equal to one of the enabled bin strings, a value corresponding to the corresponding bin string may be derived as a value of the corresponding syntax element. If not, the above-described process may be performed again after further parsing the next bit in the bitstream. Through such a process, the corresponding information may be signaled using a variable length bit even without using a start bit or an end bit for specific information (specific syntax element) in the bitstream. Through this, a relatively smaller number of bits may be allocated with respect to a low value, and thus the overall coding efficiency can be enhanced.

[0103] The decoding apparatus may perform context-based or bypass-based decoding of the respective bins in the bin string from the bitstream based on the entropy coding technique, such as CABAC or CAVLC. Here, the bitstream may include various kinds of information for image/video decoding as described above. The bitstream may be transferred to the decoding apparatus through a (digital) storage medium or a network as described above.

[0104] In the present document, in order to represent signaling of information from the encoding apparatus to the decoding apparatus, a table (syntax table) including syntax elements may be used. The order of syntax elements in the syntax table used in the present document may represent a parsing order of syntax elements from the bitstream. The encoding apparatus may configure and encode the syntax table so that the syntax elements can be parsed by the decoding apparatus in the parsing order, and the decoding apparatus may obtain values of the syntax elements by parsing and decoding the syntax elements in the corresponding syntax table in the parsing order from the bitstream.

[0105] Meanwhile, as described above, the residual samples may be derived as quantized transform coefficients through the transform and quantization processes. The quantized transform coefficients may be called transform coefficients. In this case, the transform coefficients in the block may be signaled in the form of residual information. The residual information may include a residual coding syntax. That is, the encoding apparatus may configure the residual coding syntax with the residual information, and may encode the configured residual coding syntax to output the encoded residual coding syntax in the form of a bitstream. The decoding apparatus may derive residual (quantized) transform coefficients by decoding the residual coding syntax from the bitstream. As described below, the residual coding syntax may include syntax elements representing whether the transform has been applied to the corresponding block, where is the position of the last effective transform coefficient in the block, whether an effective transform coefficient is present in a subblock, and what is the size/sign of the effective transform coefficient.

[0106] For example, the (quantized) transform coefficients may be encoded and/or decoded based on the syntax elements, such as last\_sig\_coeff\_x\_prefix, last\_sig\_coeff\_y\_prefix, last\_sig\_coeff\_x\_suffix, last\_sig\_coeff\_y\_suffix, coded\_sub\_block\_flag, sig\_coeff\_flag, par\_level\_flag, abs\_level\_gtx\_flag, abs\_remainder, coeff\_sign\_flag, dec\_abs\_level included in the residual information. This may be called residual (data) coding or (transform) coefficient coding. In this case, the transform/quantization process may be omitted. In this case, values of the residual samples may be coded and signaled in accordance with a determined method. The syntax elements related to the residual data encoding/decoding may be represented as in Table 1 below.

[Table 1]

	residual_coding( x0, y0, log2TbWidth, log2TbHeight, cldx){	Descriptor
	if((tu_mts_idx[ x0 ][ y0 ] > 0    (cu_sbt_flag && log2TbWidth < 6 && log2TbHeight < 6)) && cldx == 0 && log2TbWidth > 4)	
45	log2ZoTbWidth = 4	
	else	
	log2ZoTbWidth = Min( log2TbWidth, 5 )	
50	MaxCcbs = 2 * ( 1 << log2TbWidth ) * ( 1 << log2TbHeight )	
	if( tu_mts_idx[ x0 ][ y0 ] > 0    ( cu_sbt_flag && log2TbWidth < 6 && log2TbHeight < 6 ) ) && cldx == 0 && log2TbHeight > 4 )	
	log2ZoTbHeight = 4	
55	else	
	log2ZoTbHeight = Min( log2TbHeight, 5 )	
	if( log2TbWidth > 0 )	

(continued)

	<b>last_sig_coeff_x_prefix</b>	ae(v)
	if( log2TbHeight > 0 )	
5	<b>last_sig_coeff_y_prefix</b>	ae(v)
	if( last_sig_coeff_x_prefix > 3 )	
	<b>last_sig_coeff_x_suffix</b>	ae(v)
10	if( last_sig_coeff_y_prefix > 3 )	
	<b>last_sig_coeff_y_suffix</b>	ae(v)
	log2TbWidth = log2ZoTbWidth	
	log2TbHeight = log2ZoTbHeight	
15	remBinsPass1 = ( ( 1 << ( log2TbWidth + log2TbHeight ) ) * 7 ) >>2	
	log2SbW = ( Min( log2TbWidth, log2TbHeight ) < 2 ? 1 : 2 )	
	log2SbH = log2SbW	
20	if( log2TbWidth + log2TbHeight > 3 ) {	
	if( log2TbWidth < 2 ) {	
	log2SbW = log2TbWidth	
	log2SbH = 4 - log2SbW	
25	} else if( log2TbHeight < 2 ) {	
	log2SbH = log2TbHeight	
	log2SbW = 4 - log2SbH	
	}	
30	}	
	numSbCoeff = 1 << ( log2SbW + log2SbH )	
	lastScanPos = numSbCoeff	
35	lastSubBlock = ( 1 << ( log2TbWidth + log2TbHeight - ( log2SbW + log2SbH ) ) ) - 1	
	do {	
	if( lastScanPos == 0 ) {	
	lastScanPos = numSbCoeff	
40	lastSubBlock--	
	}	
	lastScanPos--	
45	xS = DiagScanOrder[ log2TbWidth - log2SbW ][ log2TbHeight - log2SbH ][ lastSubBlock ][ 0 ]	
	yS = DiagScanOrder[ log2TbWidth - log2SbW ][ log2TbHeight - log2SbH ][ lastSubBlock ][ 1 ]	
	xC = ( xS << log2SbW ) + DiagScanOrder[ log2SbW ][ log2SbH ][ lastScanPos ][ 0 ]	
	yC = ( yS << log2SbH ) + DiagScanOrder[ log2SbW ][ log2SbH ][ lastScanPos ][ 1 ]	
50	} while( ( xC != LastSignificantCoeffX )    ( yC != LastSignificantCoeffY ) )	
	if( lastSubBlock == 0 && log2TbWidth >= 2 && log2TbHeight >= 2 && !transform_skip_flag [ x0 ][ y0 ] && lastScanPos > 0 )	
	LfnstDcOnly = 0	
55	if( ( lastSubBlock > 0 && log2TbWidth >= 2 && log2TbHeight >= 2 )    ( lastScanPos > 7 && ( log2TbWidth == 2    log2TbWidth == 3 ) && log2TbWidth == log2TbHeight ) )	
	LfnstZeroOutSigCoeffFlag = 0	

(continued)

	QState = 0	
	for( i = lastSubBlock; i >= 0; i - - ) {	
5	startQStateSb = QState	
	xS = DiagScanOrder[ log2TbWidth - log2SbW ][ log2TbHeight - log2SbH ]	
	[ i ][ 0 ]	
10	yS = DiagScanOrder[ log2TbWidth - log2SbW ][ log2TbHeight - log2SbH ]	
	[ i ][ 1 ]	
	inferSbDcSigCoeffFlag = 0	
	if( ( i < lastSubBlock ) && ( i > 0 ) ) {	
15	<b>coded_sub_block_flag</b> [ xS ][ yS ]	ae(v)
	inferSbDcSigCoeffFlag = 1	
	}	
	firstSigScanPosSb = numSbCoeff	
20	lastSigScanPosSb = -1	
	firstPosMode0 = ( i == lastSubBlock ? lastScanPos : numSbCoeff - 1 )	
	firstPosMode1 = -1	
25	for( n = firstPosMode0; n >= 0 && remBinsPass1 >= 4; n - - ) {	
	xC = ( xS << log2SbW ) + DiagScanOrder[ log2SbW ][ log2SbH ][ n ][ 0 ]	
	yC = ( yS << log2SbH ) + DiagScanOrder[ log2SbW ][ log2SbH ][ n ][ 1 ]	
30	if( coded_sub_block_flag[ xS ][ yS ] && ( n > 0    !inferSbDcSigCoeffFlag ) && ( xC != LastSignificantCoeffX    yC != LastSignificantCoeffY ) ) {	
	<b>sig_coeff_flag</b> [ xC ][ yC ]	ae(v)
	remBinsPass1 - -	
	if( sig_coeff_flag[ xC ][ yC ] )	
35	inferSbDcSigCoeffFlag = 0	
	}	
	if( sig_coeff_flag[ xC ][ yC ] ) {	
40	<b>abs_level_gtx_flag</b> [ n ][ 0 ]	ae(v)
	remBinsPass1 - -	
	if( abs_level_gtx_flag[ n ][ 0 ] ) {	
	<b>par_level_flag</b> [ n ]	ae(v)
45	remBinsPass1 - -	
	<b>abs_level_gtx_flag</b> [ n ][ 1 ]	ae(v)
	remBinsPass1 - -	
	}	
50	if( lastSigScanPosSb == -1 )	
	lastSigScanPosSb = n	
	firstSigScanPosSb = n	
55	}	
	AbsLevelPass1[ xC ][ yC ] = sig_coeff_flag[ xC ][ yC ] + par_level_flag[ n ]	
	+ abs_level_gtx_flag[ n ][ 0 ] + 2 * abs_level_gtx_flag[ n ][ 1 ]	
	if( dep_quant_enabled_flag )	

EP 4 040 792 B9

(continued)

	QState = QStateTransTable[ QState ][ AbsLevelPass1[ xC ][ yC ] & 1 ]	
	if( remBinsPass1 < 4 )	
5	firstPosMode1 = n - 1	
	}	
	for( n = numSbCoeff - 1; n >= firstPosMode1; n-- ) {	
10	xC = ( xS << log2SbW ) + DiagScanOrder[ log2SbW ][ log2SbH ][ n ][ 0 ]	
	yC = ( yS << log2SbH ) + DiagScanOrder[ log2SbW ][ log2SbH ][ n ][ 1 ]	
	if( abs_level_gtx_flag[ n ][ 1 ] )	
	<b>abs_remainder[ n ]</b>	ae(v)
15	AbsLevel[ xC ][ yC ] = AbsLevelPass1[ xC ][ yC ] + 2 * abs_remainder[ n ]	
	}	
	for( n = firstPosMode1; n >= 0; n-- ) {	
20	xC = ( xS << log2SbW ) + DiagScanOrder[ log2SbW ][ log2SbH ][ n ][ 0 ]	
	yC = ( yS << log2SbH ) + DiagScanOrder[ log2SbW ][ log2SbH ][ n ][ 1 ]	
	<b>dec_abs_level[ n ]</b>	ac(v)
	if( AbsLevel[ xC ][ yC ] > 0 )	
25	firstSigScanPosSb = n	
	if( dep_quant_enabled_flag )	
	QState = QStateTransTable[ QState ][ AbsLevel[ xC ][ yC ] & 1 ]	
	}	
30	if( dep_quant_enabled_flag    !sign_data_hiding_enabled_flag )	
	signHidden = 0	
	else	
35	signHidden = ( lastSigScanPosSb - firstSigScanPosSb > 3 ? 1 : 0 )	
	for( n = numSbCoeff - 1; n >= 0; n-- ) {	
	xC = ( xS << log2SbW ) + DiagScanOrder[ log2SbW ][ log2SbH ][ n ][ 0 ]	
	yC = ( yS << log2SbH ) + DiagScanOrder[ log2SbW ][ log2SbH ][ n ][ 1 ]	
40	if( ( AbsLevel[ xC ][ yC ] > 0 ) && ( !signHidden    ( n != firstSigScanPosSb ) ) )	
	<b>coeff_sign_flag[ n ]</b>	ae(v)
	}	
45	if( dep_quant_enabled_flag ) {	
	QState = startQStateSb	
	for( n = numSbCoeff - 1; n >= 0; n-- ) {	
	xC = ( xS << log2SbW ) + DiagScanOrder[ log2SbW ][ log2SbH ][ n ][ 0 ]	
50	yC = ( yS << log2SbH ) + DiagScanOrder[ log2SbW ][ log2SbH ][ n ][ 1 ]	
	if( AbsLevel[ xC ][ yC ] > 0 )	
	TransCoeffLevel[ x0 ][ y0 ][ cldx ][ xC ][ yC ] = ( 2 * AbsLevel[ xC ][ yC ] - ( QState > 1 ? 1 : 0 ) ) * ( 1 - 2 * coeff_sign_flag[ n ] )	
55	QState = QStateTransTable[ QState ][ par_level_flag[ n ] ]	
	} else {	
	sumAbsLevel = 0	

(continued)

	for( n = numSbCoeff - 1; n >= 0; n-- ) {	
	xC = ( xS << log2SbW ) + DiagScanOrder[ log2SbW ][ log2SbH ][ n ][ 0 ]	
5	yC = ( yS << log2SbH ) + DiagScanOrder[ log2SbW ][ log2SbH ][ n ][ 1 ]	
	if( AbsLevel[ xC ][ yC ] > 0 ) {	
	TransCoeffLevel[ x0 ][ y0 ][ cldx ][ xC ][ yC ] = AbsLevel[ xC ][ yC ] * ( 1	
10	- 2 * coeff_sign_flag[ n ] )	
	if( signHidden ) {	
	sumAbsLevel += AbsLevel[ xC ][ yC ]	
	if( ( n == firstSigScanPosSb ) && ( sumAbsLevel % 2 ) == 1 )	
15	TransCoeffLevel[ x0 ][ y0 ][ cldx ][ xC ][ yC ] = - TransCoeffLe-	
	vel[ x0 ][ y0 ][ cldx ][ xC ][ yC ]	
	}	
	}	
20	}	
	}	
	}	
25	}	

[0107] Referring to Table 1, last\_sig\_coeff\_x\_prefix, last\_sig\_coeff\_y\_prefix, last\_sig\_coeff\_x\_suffix, and last\_sig\_coeff\_y\_suffix are syntax elements for encoding (x, y) position information of the last coefficient that is not 0 in an associated block. The associated block may be a coding block (CB) or a transform block (TB). In relation to the transform (and quantization) and residual coding process, the CB and the TB may be used interchangeably. For example, residual samples may be derived for the CB, and the (quantized) transform coefficients may be derived through the transform and quantization with respect to the residual samples as described above, and information (or syntax elements) efficiently representing (the position, size, and sign of) the (quantized) transform coefficients may be generated and signaled through the residual coding process. The quantized transform coefficients may be simply called transform coefficients. In general, if the CB is not larger than the maximum TB, the size of the CB may be equal to the size of the TB, and in this case, the target block that is transformed (and quantized) and residual-coded may be called the CB or TB. Further, if the CB is larger than the maximum TB, the target block that is transformed (and quantized) and residual-coded may be called the TB. Hereinafter, although it is explained that the syntax elements related to the residual coding are signaled in the unit of a transform block (TB), this is exemplary, and the TB may be interchangeably used with the coding block (CB) as described above.

[0108] Meanwhile, different residual coding methods may be applied depending on whether a transform skip is applied for the residual coding. As an embodiment, whether the transform skip is applied may be represented using the transform skip flag syntax element, and the residual coding may be branched in accordance with the value of the syntax element transform\_skip\_flag of the transform skip flag. That is, different syntax elements may be used for the residual coding based on the value of the transform skip flag (based on whether the transform skip is applied). The residual coding being used in case that the transform skip is not applied (i.e., in case that the transform is applied) may be called regular residual coding (RRC), and the residual coding in case that the transform skip is applied (i.e., in case that the transform is not applied) may be called transform skip residual coding (TSRC).

[0109] Table 2 below represents a process in which the residual coding is branched based on the syntax element of the transform skip flag.

[Table 2]

	transform_unit( x0, y0, tbWidth, tbHeight, treeType, subTulIndex, chType ) {	<b>Descriptor</b>
55	<b>transform_skip_flag</b> [ x0 ][ y0 ][ 0 ]	ae(v)
	if( !transform_skip_flag[ x0 ][ y0 ][ 0 ]    slice_ts_residual_coding_disabled_flag )	
	residual_coding( x0, y0, Log2( tbWidth ), Log2( tbHeight ), 0 )	

(continued)

	else	
	residual_ts_coding( x0, y0, Log2( tbWidth ), Log2( tbHeight ), 0 )	
5	}	
	if( tu_cbf_cb[ xC ][ yC ] && treeType != DUAL_TREE_LUMA ) {	
	if( sps_transform_skip_enabled_flag && !BdpcmFlag[ x0 ][ y0 ][ 1 ] && wC <= MaxTs-Size && hC <= MaxTsSize && !cu_sbt_flag )	
10	<b>transform_skip_flag[ xC ][ yC ][ 1 ]</b>	ae(v)
	if( !transform_skip_flag[ xC ][ yC ][ 1 ]    slice_ts_residual_coding_disabled_flag )	
	residual_coding( xC, yC, Log2( wC ), Log2( hC ), 1 )	
15	else	
	residual_ts_coding( xC, yC, Log2( wC ), Log2( hC ), 1 )	
	}	
20	if( tu_cbf_cr[ xC ][ yC ] && treeType != DUAL_TREE_LUMA && !( tu_cbf_cb[ xC ][ yC ] && tu_joint_cbr_residual_flag[ xC ][ yC ] ) ) {	
	if( sps_transform_skip_enabled_flag && !BdpcmFlag[ x0 ][ y0 ][ 2 ] && wC <= MaxTs-Size && hC <= MaxTsSize && !cu_sbt_flag )	
25	<b>transform_skip_flag[ xC ][ yC ][ 2 ]</b>	ae(v)
	if( transform_skip_flag[ xC ][ yC ][ 2 ]    slice_ts_residual_coding_disabled_flag )	
	residual_coding( xC, yC, Log2( wC ), Log2( hC ), 2 )	
	else	
30	residual_ts_coding( xC, yC, Log2( wC ), Log2( hC ), 2 )	
	}	
	}	

35 **[0110]** Referring to Table 2 above, in case that the transform skip is not applied (e.g., in case that the value of the transform\_skip\_flag is 0), the regular residual coding is performed, and this may be performed based on the syntax elements disclosed in Table 1 described above. Further, in case that the transform skip is applied (e.g., in case that the value of the transform\_skip\_flag is 1), the transform skip residual coding is performed, and this may be performed based on the syntax elements disclosed in Table 3 below.

40 **[0111]** Table 3 below represents the syntax elements for the transform skip residual coding.

[Table 3]

		Descriptor
	residual_ts_coding( x0, y0, log2TbWidth, log2TbHeight, cldx ) {	
45	log2SbSize = ( Min( log2TbWidth, log2TbHeight ) < 2 ? 1 : 2 )	
	numSbCoeff = 1 << ( log2SbSize << 1 )	
	lastSubBlock = ( 1 << ( log2TbWidth + log2TbHeight - 2 * log2SbSize ) ) - 1	
	inferSbCbf = 1	
50	MaxCcbS = 2 * ( 1 << log2TbWidth ) * ( 1 << log2TbHeight )	
	for( i =0; i <= lastSubBlock; i++ ) {	
	xS = DiagScanOrder[log2TbWidth-log2SbSize][log2TbHeight-log2SbSize][i][0]	
55	yS = DiagScanOrder[log2TbWidth-log2SbSize][log2TbHeight-log2SbSize][i][1]	
	if( ( i != lastSubBlock    !inferSbCbf ) {	
	<b>coded_sub_block_flag[ xS ][ yS ]</b>	ae(v)

(continued)

	}	
	if( coded_sub_block_flag[ xS ][ yS ] && i < lastSubBlock )	
5	inferSbCbf = 0	
	/* First scan pass */	
	inferSbSigCoeffFlag = 1	
10	for( n = 0; n <= numSbCoeff - 1; n++ ) {	
	xC = (xS << log2SbSize) + DiagScanOrder[ log2SbSize ][ log2SbSize ][n][ 0 ]	
	yC = (yS << log2SbSize) + DiagScanOrder[ log2SbSize ][ log2SbSize ][ n ][ 1 ]	
	if( coded_sub_block_flag[ xS ][ yS ] && ( n != numSbCoeff - 1    !inferSbSigCoeffFlag ) ) {	
15	<b>sig_coeff_flag</b> [ xC ][ yC ]	ae(v)
	MaxCcbs- -	
	if( sig_coeff_flag[ xC ][ yC ] )	
20	inferSbSigCoeffFlag = 0	
	}	
	CoeffSignLevel[ xC ][ yC ] = 0	
	if( sig_coeff_flag[ xC ][ yC ] {	
25	<b>coeff_sign_flag</b> [ n ]	ae(v)
	MaxCcbs- -	
	CoeffSignLevel[ xC ][ yC ] = ( coeff_sign_flag[ n ] > 0 ? -1 : 1 )	
30	<b>abs_level_gtx_flag</b> [ n ][ 0 ]	ae(v)
	MaxCcbs- -	
	if( abs_level_gtx_flag[ n ][ 0 ] ) {	
	<b>par_level_flag</b> [ n ]	ae(v)
35	MaxCcbs- -	
	}	
	}	
40	AbsLevelPassX[ xC ][ yC ] = sig_coeff_flag[ xC ][ yC ] + par_level_flag[ n ] + abs_level_gtx_flag [ n ][ 0 ]	
	}	
	/* Greater than X scan pass (numGtXFlags=5) */	
45	for( n = 0; n <= numSbCoeff - 1; n++ ) {	
	xC = (xS << log2SbSize) + DiagScanOrder[ log2SbSize ][ log2SbSize ][ n ][ 0 ]	
	yC = (yS << log2SbSize) + DiagScanOrder[ log2SbSize ][ log2SbSize ][ n ][ 1 ]	
	for( j = 1; j < 5; j++ ) {	
50	if( abs_level_gtx_flag[ n ][ j - 1 ] )	
	<b>abs_level_gtx_flag</b> [ n ][ j ]	ae(v)
	MaxCcbs- -	
55	AbsLevelPassX[ xC ][ yC ] += 2 * abs_level_gtx_flag[ n ][ j ]	
	}	
	}	
	/* remainder scan pass */	

(continued)

	for( n = 0; n <= numSbCoeff - 1; n++ ) {	
	xC = (xS << log2SbSize ) + DiagScanOrder[ log2SbSize ][ log2SbSize ][ n ][ 0 ]	
5	yC = (yS << log2SbSize ) + DiagScanOrder[ log2SbSize ][ log2SbSize ][ n ][ 1 ]	
	if( abs_level_gtx_flag[ n ][ 4 ] )	
	<b>abs_remainder[ n ]</b>	ae(v)
10	if(intra_bdpcm_flag = 0) {	
	absRightCoeff = abs( TransCoeffLevel[ x0 ][ y0 ][ cldx ][ xC - 1 ][ yC ] )	
	absBelowCoeff = abs(TransCoeffLevel[ x0 ][ y0 ][ cldx ][ xC ][ yC - 1 ] )	
	predCoeff = Max( absRightCoeff, absBelowCoeff )	
15	if( AbsLevelPassX[ xC ][ yC ] + abs_remainder[ n ] = 1 && predCoeff > 0)	
	TransCoeffLevel[ x0 ][ y0 ][ cldx ][ xC ][ yC ] = ( 1 - 2 * coeff_sign_flag[ n ] ) * predCoeff	
	else if( AbsLevelPassX[ xC ][ yC ] + abs_remainder[ n ] <= predCoeff)	
20	TransCoeffLevel[ x0 ][ y0 ][ cldx ][ xC ][ yC ] = ( 1 - 2 * coeff_sign_flag[ n ] )	
	* ( AbsLevelPassX[ xC ][ yC ] + abs_remainder[ n ] - 1 )	
	else	
25	TransCoeffLevel[ x0 ][ y0 ][ cldx ][ xC ][ yC ] = ( 1 - 2 * coeff_sign_flag[ n ] )	
	* ( AbsLevelPassX[ xC ][ yC ] + abs_remainder[ n ] )	
	} else	
30	TransCoeffLevel[ x0 ][ y0 ][ cldx ][ xC ][ yC ] = ( 1 - 2 * coeff_sign_flag[ n ] )	
	* ( AbsLevelPassX[ xC ][ yC ] + abs remainder[ n ] )	
	}	
	}	
	}	

35 **[0112]** For example, the transform skip flag indicating whether the transform skip of the transform block is performed may be parsed, and it may be determined whether the transform skip flag is 1. In case that the value of the transform skip flag is 1, as shown in Table 3, syntax elements sig\_coeff\_flag, coeff\_sign\_flag, abs\_level\_gtx\_flag, par\_level\_flag, and/or abs\_remainder for the residual coefficient of the transform block may be parsed, and the residual coefficient may be derived based on the syntax elements. In this case, the syntax elements may be sequentially parsed, and the parsing order may be changed. Here, the abs\_level\_gtx\_flag may represent abs\_level\_gt1\_flag, abs\_level\_gt3\_flag, abs\_level\_gt5\_flag, abs\_level\_gt7\_flag, and/or abs\_level\_gt9\_flag. For example, the abs\_level\_gtx\_flag[n][j] may be a flag representing whether an absolute value of the transform coefficient level (or value obtained by shifting the transform coefficient level by 1 to the right) is larger than (j<<1)+1 at a scanning position n. The (j<<1)+1 may be replaced by a specific threshold value, such as a first threshold value or a second threshold value in some cases.

45 **[0113]** Further, in case that the value of the transform skip flag is 0, as shown in Table 1, syntax elements sig\_coeff\_flag, abs\_level\_gtx\_flag, par\_level\_flag, abs\_remainder, dec\_abs\_level, and coeff\_sign\_flag for the residual coefficient of the transform block may be parsed, and the residual coefficient may be derived based on the syntax elements. In this case, the syntax elements may be sequentially parsed, and the parsing order may be changed. Here, the abs\_level\_gtx\_flag may represent abs\_level\_gt1\_flag and/or abs\_level\_gt3\_flag.

50 **[0114]** Meanwhile, as described above, the encoding apparatus may derive a residual block (residual samples) based on a predicted block (prediction samples) through intra/inter/IBC/palette predictions, and may derive quantized transform coefficients by applying transform and quantization with respect to the derived residual samples. The information (residual information) on the quantized transform coefficients may be included in the residual coding syntax, and may be output in the form of a bitstream after being encoded. The decoding apparatus may obtain information (residual information) on the quantized transform coefficients from the bitstream, and may derive the quantized transform coefficients by decoding the obtained information. The decoding apparatus may derive the residual samples through dequantization/inverse transform based on the quantized transform coefficients. As described above, at least one of the quantization/dequantization and/or transform/inverse transform may be omitted. In case that the transform/inverse transform is omitted, the transform

coefficient may be called a coefficient or residual coefficient, or for unity of expression, may still be called the transform coefficient. Whether to omit the transform/inverse transform may be signaled based on the transform `_skip_flag`. For example, if the value of the transform `_skip_flag` is 1, it may represent that the transform/inverse transform is omitted, and this may be referred to as a transform skip mode.

5 **[0115]** In general, in the video/image coding, the quantization rate may be changed, and the compression rate may be adjusted using the changed quantization rate. From the viewpoint of implementation, a quantization parameter (QP) may be used instead of direct usage of the quantization rate in consideration of complexity. For example, a quantization parameter of an integer value in the range of 0 to 63 may be used, and each value of the quantization parameter may correspond to the actual quantization rate. Further, for example, a quantization parameter QP<sub>y</sub> for a luma component  
10 (luma sample) and a quantization parameter QP<sub>c</sub> for a chroma component (chroma sample) may be differently configured.

**[0116]** In a quantization process, the quantized transform coefficient C' may be obtained by dividing an input transform coefficient C by the quantization rate Q<sub>step</sub>. In this case, in consideration of computational complexity, the quantization rate is made in an integer form by multiplying the quantization rate by a scale, and a shift operation may be performed as much as the value corresponding to the scale value. A quantization scale may be derived based on a product of the quantization rate and the scale value. That is, the quantization scale may be derived according to the QP. For example, the quantization scale may be applied to the transform coefficient C, and based on this, the quantized transform coefficient C' may be derived.

**[0117]** The dequantization process is reverse to the quantization process, and the quantized transform coefficient C' may be multiplied by the quantization rate Q<sub>step</sub>, and based on this, a reconstructed transform coefficient C" may be  
20 obtained. In this case, a level scale may be derived according to the quantization parameter, the level scale may be applied to the quantize3d transform coefficient C', and based on this, the reconstructed transform coefficient C" may be derived. The reconstructed transform coefficient C" is somewhat different from the initial transform coefficient C due to a loss in the transform and/or quantization process. Accordingly, even the encoding apparatus performs dequantization in the same manner as in the decoding apparatus.

25 **[0118]** Further, in performing the prediction, it may be based on palette coding. The palette coding is a useful technology to represent blocks including a small number of unique color values. Instead of applying the prediction and transform to the block, in a palette mode, an index is signaled to represent the value of each sample. The palette mode is useful to save a video memory buffer space. The block may be coded using the palette mode (e.g., `MODE_PLT`). In order to decode the encoded block as described above, the decoder should decode a pallet entry and an index. The palette entry may be  
30 represented by a palette table, and may be encoded by a palette table coding tool.

**[0119]** The palette coding may be called an (intra) palette mode or an (intra) palette coding mode. The current block may be reconstructed according to the palette coding or palette mode. The palette coding may be seen as an example of intra coding, or may be seen as one of intra prediction methods. However, in a similar manner to the above-described skip mode, a separate residual value for the corresponding block may not be signaled.

35 **[0120]** For example, in case that the palette mode is selected, information on a palette table may be signaled. The palette table may include an index corresponding to each pixel. The palette table may configure a palette prediction table from pixel values used in the previous block. For example, previously used pixel values may be stored in a specific buffer (palette predictor), and palette predictor information `palette_predictor_run` for configuring the current palette may be received from the buffer. That is, the palette predictor may include data representing an index for at least a part of a palette  
40 index map of the current block. In case that the palette entry for expressing the current block is not enough as a palette prediction entry configured from the palette predictor, pixel information on the current palette entry may be separately transmitted.

**[0121]** The palette mode may be signaled at a CU level, and may be generally used in case that most pixels in the CU may be represented as a set of representative pixel values. That is, in the palette mode, samples in the CU may be  
45 expressed as a set of representative pixel values. Such a set may be referred to as a palette. In case of a sample having a value close to the pixel value in the palette, a palette index `palette_idx_idc` corresponding to the pixel value in the palette or information capable of indicating the index (`run_copy_flag`, `copy_above_palette_indices_flag`) may be signaled. In case of a sample having a pixel value excluding the palette entry, the sample may be indicated as an escape symbol, and a quantized sample value may be directly signaled. In the present document, the pixel or the pixel value may be referred to as  
50 the sample or the sample value.

**[0122]** In order to decode a block coded in a palette mode, the decoder requires palette entry information and palette index information. In case that the palette index corresponds to the escape symbol, a (quantized) escape value may be signaled as an additional component. Further, the encoder should derive a proper palette for the corresponding CU, and transfer the palette to the decoder.

55 **[0123]** For efficient coding of the pallet entry, a palette predictor may be maintained. The palette predictor and the maximum size of the palette may be signaled through SPS. Further, the palette predictor and the maximum size of the palette may be predefined. For example, the palette predictor and the maximum size of the palette may be defined as 31 and 15, respectively, depending on whether the current block is a single tree or a dual-tree. In VVC standards,

sps\_palette\_enabled\_flag representing whether the palette mode is enabled may be transmitted. Then, a pred\_mod\_e\_plt\_coding flag representing whether the current coding unit is coded in the palette mode may be transmitted. The palette predictor may be initialized at the beginning of each brick or each slice.

[0124] With respect to each entry in the palette predictor, a reuse flag may be signaled, and may represent whether the entry is a part of the current pallet. The reuse flag may be transmitted using run-length coding of 0. Thereafter, the number of new palette entries may be signaled using the 0<sup>th</sup>-order exponential Golomb coding. Last, a component value for the new palette entry may be signaled. After the current CU is encoded, the palette predictor may be updated using the current palette, and the entry of the previous palette predictor that is not reused in the current palette may be added to the end of the new palette predictor (palette stuffing) until it reaches the allowable maximum size.

[0125] In order to code a palette index map, the index may be coded using horizontal and vertical traverse scans. The scan order may be explicitly signaled from the bitstream using the flag information (e.g., palette\_transpose\_flag).

[0126] Meanwhile, the palette index may be coded using two kinds of palette sample modes, and for example, "INDEX" mode and "COPY\_ABOVE" mode may be used. Such palette modes may be signaled using a flag representing whether the palette mode is the "INDEX" mode or the "COPY\_ABOVE" mode. In this case, the escape symbol may be signaled in the "INDEX" mode, and an index having the same size as the current palette size may be allocated. For example, if it is assumed that the size of the current palette is 10, No. 0 to No. 9 indexes may mean entry indexes in the palette, and No. 10 index may mean an index for the escape symbol. In case that the horizontal scan is used, the flag may be signaled excluding the top row, and in case that the vertical scan is used, or the previous mode is the "COPY\_ABOVE" mode, the flag may be signaled excluding the first column. In the "COPY\_ABOVE" mode, the palette index of the sample of the row on the top may be copied. In the "INDEX" mode, the palette index may be explicitly signaled. In both the "INDEX" mode and the "COPY\_ABOVE" mode, the run value representing the number of next samples being coded using the same mode may be signaled. In case that the escape symbol is a part of the run in the "INDEX" mode or the "COPY\_ABOVE" mode, escape component values may be signaled with respect to each escape symbol.

[0127] The coding of the palette index may be as follows. First, the number of indexes for the CU may be signaled. Next, actual indexes for the whole CU may be signaled using fixed length coding. The number of indexes and the index may be coded in the bypass mode. Through this, index-related bypass bins may be grouped together. Next, the palette sample mode copy\_above\_palette\_indices\_flag and the run may be signaled in an interleaved manner. Last, component escape values corresponding to the escape samples for the whole CU may be grouped together, and may be coded in the bypass mode.

[0128] Table 4 below represents an example of a syntax structure including syntax elements related to the palette mode based coding for the coding unit, and Table 5 below represents semantics for syntax elements included in the syntax of Table 4.

[Table 4]

		Descriptor
35	palette_coding( x0, y0, cbWidth, cbHeight, treeType ) {	
	startComp = ( treeType == DUAL_TREE_CHROMA ) ? 1 : 0	
40	numComps = ( treeType == SINGLE_TREE ) ? ( sps_chroma_format_idc == 0 ? 1 : 3 ) : ( treeType == DUAL_TREE_CHROMA ) ? 2 : 1	
	maxNumPaletteEntries = ( treeType == SINGLE_TREE ) ? 31 : 15	
	palettePredictionFinished = 0	
	NumPredictedPaletteEntries = 0	
45	for( predictorEntryIdx = 0; predictorEntryIdx < PredictorPaletteSize[ startComp ] && !palettePredictionFinished && NumPredictedPaletteEntries < maxNumPaletteEntries; predictorEntryIdx++ ) {	
	<b>palette_predictor_run</b>	ae(v)
50	if( palette_predictor_run != 1 ) {	
	if( palette_predictor_run > 1 )	
	predictorEntryIdx += palette_predictor_run - 1	
	PalettePredictorEntryReuseFlags[ predictorEntryIdx ] = 1	
55	NumPredictedPaletteEntries++	
	} else	
	palettePredictionFinished = 1	

(continued)

	}	
	if( NumPredictedPaletteEntries < maxNumPaletteEntries )	
5	<b>num_signalled_palette_entries</b>	ae(v)
	for( cldx = startComp; cldx < ( startComp + numComps ); cldx++ )	
	for( i = 0; i < num_signalled_palette_entries; i++ )	
10	<b>new_palette_entries[ cldx ][ i ]</b>	ae(v)
	if( CurrentPaletteSize[ startComp ] > 0 )	
	<b>palette_escape_val_present_flag</b>	ae(v)
	if( MaxPaletteIndex > 0 ) {	
15	adjust = 0	
	<b>palette_transpose_flag</b>	ae(v)
	}	
20	if( treeType != DUAL_TREE_CHROMA && palette_escape_val_present_flag )	
	if( pps_cu_qp_delta_enabled_flag && !IsCuQpDeltaCoded ) {	
	<b>cu_qp_delta_abs</b>	ae(v)
	if( cu_qp_delta_abs )	
25	<b>cu_qp_delta_sign_flag</b>	ae(v)
	}	
	if( treeType != DUAL_TREE_LUMA && palette_escape_val_present_flag )	
30	if( sh_cu_chroma_qp_offset_enabled_flag && !IsCuChromaQpOffsetCoded ) {	
	<b>cu_chroma_qp_offset flag</b>	ae(v)
	if( cu_chroma_qp_offset_flag && pps_chroma_qp_offset_list_len_minus1 > 0 )	
	<b>cu_chroma_qp_offset_idx</b>	ae(v)
35	}	
	PreviousRunPosition = 0	
	PreviousRunType = 0	
40	for( subSetId = 0; subSetId <= ( cbWidth * cbHeight - 1 ) / 16; subSetId++ ) {	
	minSubPos = subSetId * 16	
	if( minSubPos + 16 > cbWidth * cbHeight )	
	maxSubPos = cbWidth * cbHeight	
45	else	
	maxSubPos = minSubPos + 16	
	RunCopyMap[ x0 ][ y0 ] = 0	
	PaletteScanPos = minSubPos	
50	log2CbWidth = Log2( cbWidth )	
	log2CbHeight = Log2( cbHeight )	
	while( PaletteScanPos < maxSubPos ) {	
55	xC = x0+TraverseScanOrder[log2CbWidth][log2CbHeight] [PaletteScanPos] [0]	
	yC = y0+TraverseScanOrder[log2CbWidth][log2CbHeight] [PaletteScanPos] [1]	
	if( PaletteScanPos > 0 ) {	

EP 4 040 792 B9

(continued)

	xcPrev = x0 + TraverseScanOrder[log2CbWidth][log2CbHeight][PaletteScanPos-1] [0]	
	ycPrev = y0 + TraverseScanOrder[log2CbWidth][log2CbHeight][PaletteScanPos-1] [ 1 ]	
5	}	
	if( MaxPaletteIndex > 0 && PaletteScanPos > 0 ) {	
	<b>run_copy_flag</b>	ae(v)
10	RunCopyMap[ xC ][ yC ] = run_copy_flag	
	}	
	CopyAboveIndicesFlag[ xC ][ yC ] = 0	
	if( MaxPaletteIndex > 0 && !RunCopyMap[ xC ][ yC ] ) {	
15	if( ( ( !palette_transpose_flag[ && yC > y0 )    ( palette_transpose_flag && xC > x0 ) ) && CopyAboveIndicesFlag[ xcPrev ][ ycPrev ] = 0 && PaletteScanPos > 0 ) {	
	<b>copy_above_palette_indices_flag</b>	ae(v)
20	CopyAboveIndicesFlag[ xC ][ yC ] = copy_above_palette_indices_flag[	
	}	
	PreviousRunType = CopyAboveIndicesFlag[ xC ][ yC ]	
	PreviousRunPosition = PaletteScanPos	
25	} else if( PaletteScanPos > 0 )	
	CopyAboveIndicesFlag[xC][yC] = CopyAboveIndicesFlag[xcPrev][ycPrev]	
	PaletteScanPos ++	
30	}	
	PaletteScanPos = minSubPos	
	while( PaletteScanPos < maxSubPos ) {	
	xC = x0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos ][ 0 ]	
35	yC = y0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ PaletteScanPos ][ 1 ]	
	if( PaletteScanPos > 0 ) {	
	xcPrev =x0 + TraverseScanOrder[log2CbWidth][log2CbHeight][PaletteScanPos-1]	
40	[0]	
	ycPrev = y0 + TraverseScanOrder[log2CbWidth][log2CbHeight][PaletteScan-	
	Pos-1][1]	
	}	
45	if( MaxPaletteIndex > 0 && !RunCopyMap[ xC ][ yC ] && CopyAboveIndicesFlag[ xC ][ yC ] = = 0 ) {	
	if( MaxPaletteIndex - adjust > 0 )	
	<b>palette_idx_idc</b>	ae(v)
50	adjust = 1	
	}	
	if( !RunCopyMap[ xC ][ yC ] && CopyAboveIndicesFlag[ xC ][ yC ] = = 0 )	
	CurrPaletteIndex = palette_idx_idc	
55	if( CopyAboveIndicesFlag[ xC ][ yC ] = = 0 )	
	PaletteIndexMap[ xC ][ yC ] = CurrPaletteIndex	
	else if( !palette_transpose_flag )	

EP 4 040 792 B9

(continued)

	PaletteIndexMap[ xC ][ yC ] = PaletteIndexMap[ xC ][ yC - 1 ]	
	else	
5	PaletteIndexMap[ xC ][ yC ] = PaletteIndexMap[ xC - 1 ][ yC ]	
	PaletteScanPos ++	
	}	
10	if( palette_escape_val_present_flag ) {	
	for( cldx = startComp; cldx < ( startComp + numComps ); cldx++ ) {	
	for( sPos = minSubPos; sPos < maxSubPos; sPos++ ) {	
	xC = x0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ sPos ][ 0 ]	
15	yC = y0 + TraverseScanOrder[ log2CbWidth ][ log2CbHeight ][ sPos ][ 1 ]	
	if( !( treeType == SINGLE_TREE && cldx != 0 && ( xC % SubWidthC != 0    yC % Sub-HeightC != 0 ) ) ) {	
	if( PaletteIndexMap[ cldx ][ xC ][ yC ] == MaxPaletteIndex ) {	
20	<b>palette_escape_val</b>	ae(v)
	PaletteEscapeVal[ cldx ][ xC ][ yC ] = palette_escape_val	
	}	
25	}	
	}	
	}	
	}	
30	}	
	}	

[Table 5]

35  
40  
45  
50  
55

5  
10  
15  
In the following semantics, the array indices  $x_0$ ,  $y_0$  specify the location (  $x_0$ ,  $y_0$  ) of the top-left luma sample of the considered coding block relative to the top-left luma sample of the picture. The array indices  $x_C$ ,  $y_C$  specify the location (  $x_C$ ,  $y_C$  ) of the sample relative to the top-left luma sample of the picture, when `treeType` is equal to `SINGLE_TREE` or `DUAL_TREE_LUMA`; and relative to the top-left chroma sample of the picture, when `treeType` is equal to `DUAL_TREE_CHROMA`. The array index `startComp` specifies the first colour component of the current palette table. `startComp` equal to 0 indicates the Y component; `startComp` equal to 1 indicates the Cb component; `startComp` equal to 2 indicates the Cr component. `numComps` specifies the number of colour components in the current palette table.

20  
The predictor palette consists of palette entries from previous coding units that are used to predict the entries in the current palette.

25  
PredictorPaletteSize[ `startComp` ] specifies the size of the predictor palette for the first colour component of the current palette table `startComp`. PredictorPaletteSize[ `startComp` ] is derived as specified in subclause 8.4.5.3.

30  
35  
PalettePredictorEntryReuseFlags[  $i$  ] equal to 1 specifies that the  $i$ -th entry in the predictor palette is reused in the current palette. PalettePredictorEntryReuseFlags[  $i$  ] equal to 0 specifies that the  $i$ -th entry in the predictor palette is not an entry in the current palette. All elements of the array PalettePredictorEntryReuseFlags[  $i$  ] are initialized to 0.

**palette\_predictor\_run** is used to determine the number of zeros that precede a non-zero entry in the array PalettePredictorEntryReuseFlags.

40  
45  
50  
55  
It is a requirement of bitstream conformance that the value of `palette_predictor_run` shall be in the range of 0 to ( PredictorPaletteSize[ `startComp` ] – `predictorEntryIdx` ), inclusive, where `predictorEntryIdx` corresponds to the current position in the array PalettePredictorEntryReuseFlags. The variable NumPredictedPaletteEntries specifies the number of entries in the current palette that are reused from the predictor palette. The value of NumPredictedPaletteEntries shall be in the range of 0 to `maxNumPaletteEntries`, inclusive.

**num\_signalled\_palette\_entries** specifies the number of entries in the current palette that are explicitly signalled for the first colour component of the current palette table startComp.

When **num\_signalled\_palette\_entries** is not present, it is inferred to be equal to 0.

The variable **CurrentPaletteSize[ startComp ]** specifies the size of the current palette for the first colour component of the current palette table startComp and is derived as follows:

$$\text{CurrentPaletteSize[ startComp ]} = \text{NumPredictedPaletteEntries} + \text{num\_signalled\_palette\_entries} \quad (176)$$

The value of **CurrentPaletteSize[ startComp ]** shall be in the range of 0 to **maxNumPaletteEntries**, inclusive.

**new\_palette\_entries[ cIdx ][ i ]** specifies the value for the i-th signalled palette entry for the colour component cIdx.

The variable **LocalDualTreeFlag** is derived as follows:

$$\text{LocalDualTreeFlag} = ( \text{treeType} \neq \text{SINGLE\_TREE} \ \&\& \ ( \text{sh\_slice\_type} \neq \text{I} \ || \ ( \text{sh\_slice\_type} = \text{I} \ \&\& \ \text{sps\_qtbtt\_dual\_tree\_intra\_flag} = 0 ) ) ) ? 1 : 0 \quad (177)$$

The variable **PredictorPaletteEntries[ cIdx ][ i ]** specifies the i-th element in the predictor palette for the colour component cIdx.

The variable **CurrentPaletteEntries[ cIdx ][ i ]** specifies the i-th element in the current palette for the colour component cIdx and is derived as follows:

```

numPredictedPaletteEntries = 0
for( i = 0; i < PredictorPaletteSize[ startComp ]; i++ )
    if( PalettePredictorEntryReuseFlags[ i ] ) {
        for( cIdx = LocalDualTreeFlag ? 0 : startComp;
            cIdx < LocalDualTreeFlag ? 3 : ( startComp + numComps ); cIdx++ )
            CurrentPaletteEntries[ cIdx ][ numPredictedPaletteEntries ] =
                PredictorPaletteEntries[ cIdx ][ i ]
    }

```

```

        numPredictedPaletteEntries++
    }
5   for( cIdx = startComp; cIdx < (startComp + numComps); cIdx++)      (178)
        for( i = 0; i < num_signalled_palette_entries; i++ )
            CurrentPaletteEntries[ cIdx ][ numPredictedPaletteEntries + i ] =
10                                     new_palette_entries[ cIdx ][ i ]

```

15 **palette\_escape\_val\_present\_flag** equal to 1 specifies that the current coding unit contains at least one escape coded sample. **palette\_escape\_val\_present\_flag** equal to 0 specifies that there are no escape coded samples in the current coding unit. When not present, the value of **palette\_escape\_val\_present\_flag** is inferred to be equal to 1.

20 The variable **MaxPaletteIndex** specifies the maximum possible value for a palette index for the current coding unit. The value of **MaxPaletteIndex** is set equal to  $\text{CurrentPaletteSize}[\text{startComp}] - 1 + \text{palette\_escape\_val\_present\_flag}$ .

25 **palette\_idx\_idc** is an indication of an index to the palette table, **CurrentPaletteEntries**. The value of **palette\_idx\_idc** shall be in the range of 0 to **MaxPaletteIndex**, inclusive, for the first index in the block and in the range of 0 to  $(\text{MaxPaletteIndex} - 1)$ , inclusive, for the remaining indices in the block.

30 When **palette\_idx\_idc** is not present, it is inferred to be equal to 0.

35 **palette\_transpose\_flag** equal to 1 specifies that vertical traverse scan is applied for scanning the indices for samples in the current coding unit. **palette\_transpose\_flag** equal to 0 specifies that horizontal traverse scan is applied for scanning the indices for samples in the current coding unit. When not present, the value of **palette\_transpose\_flag** is inferred to be equal to 0.

40 The array **TraverseScanOrder** specifies the scan order array for palette coding. If **palette\_transpose\_flag** is equal to 0, **TraverseScanOrder** is assigned the horizontal scan order **HorTravScanOrder**. Otherwise (**palette\_transpose\_flag** is equal to 1), **TraverseScanOrder** is assigned the vertical scan order **VerTravScanOrder**.

45 **run\_copy\_flag** equal to 1 specifies that the palette run type is the same as the run type at the previously scanned position and palette index is the same as the index at the previous scanned position if **CopyAboveIndicesFlag**[ xC ][ yC ] is equal to 0.

Otherwise, run\_copy\_flag equal to 0 specifies that the palette run type is different from the run type at the previously scanned position.

**copy\_above\_palette\_indices\_flag** equal to 1 specifies that the palette index is equal to the palette index at the same location in the row above if horizontal traverse scan is used or the same location in the left column if vertical traverse scan is used. copy\_above\_palette\_indices\_flag equal to 0 specifies that an indication of the palette index of the sample is coded in the bitstream or inferred.

The variable CopyAboveIndicesFlag[ xC ][ yC ] equal to 1 specifies that the palette index is copied from the palette index in the row above (horizontal scan) or left column (vertical scan). CopyAboveIndicesFlag[ xC ][ yC ] equal to 0 specifies that the palette index is explicitly coded in the bitstream or inferred.

The variable PaletteIndexMap[ xC ][ yC ] specifies a palette index, which is an index to the array represented by CurrentPaletteEntries. The value of PaletteIndexMap[ xC ][ yC ] shall be in the range of 0 to MaxPaletteIndex, inclusive.

The variable adjustedRefPaletteIndex is derived as follows:

```

adjustedRefPaletteIndex = MaxPaletteIndex + 1
if( PaletteScanPos > 0 ) {
    xcPrev = x0 +
        TraverseScanOrder[log2CbWidth][log2bHeight][PaletteScanPos-1][0]
    ycPrev = y0 +
        TraverseScanOrder[log2CbWidth][log2bHeight][PaletteScanPos-1][1]
    if( CopyAboveIndicesFlag[ xcPrev ][ ycPrev ] == 0 )
        adjustedRefPaletteIndex = PaletteIndexMap[xcPrev][ycPrev]    (179)
    else {
        if( !palette_transpose_flag )
            adjustedRefPaletteIndex = PaletteIndexMap[ xC ][ yC - 1 ]
        else
            adjustedRefPaletteIndex = PaletteIndexMap[ xC - 1 ][ yC ]
    }
}

```

When CopyAboveIndicesFlag[ xC ][ yC ] is equal to 0, the variable CurrPaletteIndex

is derived as follows:

```

5         if( CurrPaletteIndex >= adjustedRefPaletteIndex )
           CurrPaletteIndex++                                     (180)

```

**palette\_escape\_val** specifies the quantized escape coded sample value for a component.

The variable `PaletteEscapeVal[ cIdx ][ xC ][ yC ]` specifies the escape value of a sample for which `PaletteIndexMap[ xC ][ yC ]` is equal to `MaxPaletteIndex` and `palette_escape_val_present_flag` is equal to 1. The array index `cIdx` specifies the colour component.

It is a requirement of bitstream conformance that `PaletteEscapeVal[ cIdx ][ xC ][ yC ]` shall be in the range of 0 to  $(1 \ll \text{BitDepth}) - 1$ , inclusive.

**[0129]** Referring to Table 4 and Table 5 above, in case that a palette mode is applied for the current block (i.e., current coding unit), the palette coding syntax (e.g., `palette_coding()`) as in Table 4 above may be parsed/signaled.

**[0130]** For example, the palette table may be configured based on the palette entry information. The palette entry information may include syntax elements, such as `palette_predictor_run`, `num_signalled_palette_entries`, and `new_palette_entries`.

**[0131]** Further, a palette index map for the current block may be configured based on the palette index information. The palette index information may include syntax elements, such as `num_palette_indices_minus1`, `palette_idx_idc`, and `palette_transpose_flag`. Based on the palette index information as described above, the palette index map (e.g., `PaletteIndexMap`) may be configured by deriving the palette index (e.g., `PaletteIndexIdc`) for the samples in the current block while circulating in accordance with a traverse scan direction (vertical direction or horizontal direction).

**[0132]** Further, based on the palette index map, a sample value for the palette entry in the palette table may be derived, and reconstructed samples of the current block may be generated based on the sample value mapped on the palette entry.

**[0133]** Further, in case that a sample having an escape value is present (i.e., in case that the value of the `palette_escape_val_present_flag` is 1) in the current block, the escape value for the current block may be derived based on the escape information. The escape information may include syntax elements, such as `palette_escape_val_present_flag` and `palette_escape_val`. For example, based on the quantized escape value information (e.g., `palette_escape_val`), the escape value for the escape-coded sample in the current block may be derived. The reconstructed samples of the current block may be generated based on the escape value.

**[0134]** Meanwhile, in the encoding/decoding process, block differential pulse coded modulation or block-based delta pulse code modulation (BDPCM) technique may be used. The BDPCM may be named quantized residual block-based delta pulse code modulation (RDPCM).

**[0135]** In case that the block is predicted by applying the BDPCM, reconstructed samples may be utilized in order to predict rows or columns of the block line by line. In this case, a used reference sample may be a non-filtered sample. The BDPCM direction may represent whether a vertical direction or horizontal direction prediction has been used. That is, in case that the BDPCM is applied, the vertical direction or the horizontal direction may be selected as the BDPCM direction, and the prediction may be performed in the BDPCM direction. A prediction error may be quantized in a spatial domain, and the sample may be reconstructed by adding the dequantized prediction error to the prediction (i.e., prediction sample). The prediction error may mean the residual. The quantized residual domain BDPCM may be proposed as an alternative of the BDPCM, and the prediction direction or signaling may be equal to the BDPCM applied to the spatial domain. That is, the residual may be reconstructed through the dequantization after the quantization coefficients themselves are built up like delta pulse code modulation (DPCM) through the quantized residual domain BDPCM. Accordingly, the quantized residual domain BDPCM may be used as the meaning that a residual coding end applies the DPCM. The quantized residual domain used hereinafter is obtained by quantizing the residual derived based on the prediction without being transformed, and means a domain for the quantized residual sample. For example, the quantized residual domain may include the quantized residual (or quantized residual coefficient) to which the transform skip is applied, that is, the transform is skipped with

respect to the residual sample, but the quantization is applied thereto. Further, for example, the quantized residual domain may include the quantized transform coefficient.

**[0136]** As described above, the BDPCM may be applied to the quantized residual domain, the quantized residual domain may include the quantized residual (or quantized residual coefficient), and in this case, the transform skip may be applied with respect to the residual. That is, in case that the BDPCM is applied, the transform may be skipped and the quantization may be applied with respect to the residual sample. Further, the quantized residual domain may include the quantized transform coefficient. A flag representing whether the BDPCM is applicable may be signaled at a sequence level (SPS), and such a flag may be signaled only in case of being signaled that the transform skip mode is possible at the SPS. The flag may be called a BDPCM enabled flag or SPS BDPCM enabled flag.

**[0137]** In case of applying the BDPCM, intra prediction may be performed with respect to entire blocks by sample copy according to a prediction direction (e.g., vertical prediction or horizontal prediction) similar to the intra prediction direction. The residual that is the difference value between the original and the prediction blocks may be quantized through skipping of the transform, and a delta value, that is, the difference value between the quantized residual and the predictor in the horizontal or vertical direction (i.e., quantized residual in the horizontal or vertical direction), may be coded.

**[0138]** If the BDPCM is applicable, the CU size may be equal to or smaller than MaxTsSize (maximum transform skip block size) for the luma sample, and in case that the CU is coded through intra prediction, the flag information may be transmitted at the CU level. The flag information may be called the BDPCM flag. Here, the MaxTsSize may mean the maximum block size for the transform skip mode to be allowed. The flag information may indicate whether typical intra coding is applied or the BDPCM is applied. If the BDPCM is applied, a BDPCM prediction direction flag indicating whether the prediction direction is the horizontal direction or the vertical direction may be transmitted. The BDPCM prediction direction flag may be called a BDPCM direction flag. Thereafter, the block may be predicted through a typical horizontal or vertical intra prediction process using a non-filtered reference sample. Further, the residual may be quantized, and the difference value between the quantized residual and the predictor, for example, between already quantized residuals in surrounding positions in the horizontal or vertical direction according to the BDPCM prediction direction may be coded.

**[0139]** Meanwhile, as described above, information (syntax element) in a syntax table disclosed in the present document may be included in image/video information, and may be configured/encoded by the encoding apparatus to be transferred to the decoding apparatus in the form of a bitstream. The decoding apparatus may parse/decode the information (syntax element) in the corresponding syntax table. The decoding apparatus may perform a decoding process (prediction, (transform skip based) residual process, BDPCM, and palette coding) for the current block based on the decoded information.

**[0140]** Hereinafter, in the present document, an efficient scheme for parsing/signaling a syntax element having dependency is proposed with respect to a transform skip related high-level syntax element and/or palette coding related high-level syntax element. That is, according to an embodiment of the present document, during the video/image coding, whether to perform coding may be classified in accordance with dependency and non-dependency of information which is absolutely necessary or auxiliary used in performing the transform skip and/or palette coding, and thus efficient coding may be performed.

**[0141]** In video coding, a switch of a coding tool may be defined in a specific high-level syntax (HLS). In case of the VVC in the related art, flag information on respective coding tools may be defined in the SPS. Further, in the VVC, standardization has been in progress towards having independence among respective high-level syntax sets (e.g., video parameter set (VPS), sequence parameter set (SPS), picture parameter set (PPS), adaptation parameter set (APS), decoding parameter set (DPS), and slice header). Accordingly, in the high-level syntax set in which a flag of a coding tool is present, a plurality of syntax elements having dependency are present. In embodiment(s) of the present document, a method for parsing/signaling a high-level syntax element having dependency in accordance with the transform skip and/or palette coding is proposed.

**[0142]** As an embodiment, the present document proposes a method for saving bits being transmitted by enabling the syntax element having dependency to determine whether to perform parsing/signaling in accordance with the dependent condition with respect to the transform skip related high-level syntax element. As an example, a method for parsing the high-level syntax element having dependency depending on whether to use the transform skip by the transform skip (enabled) flag is proposed.

**[0143]** For example, as syntax elements dependent on the transform skip based coding, there are a transform skip (enabled) flag (e.g., `sps_transform_skip_enabled_flag`), minimum quantization parameter information for transform skip (e.g., `min_qp_prime_ts_minus4`), and information on whether to apply the BDPCM (e.g., `sps_bdpcm_enabled_flag`). As an example, if the value of the transform skip (enabled) flag is defined as 1, the related flag or information syntax elements should be necessarily transmitted, whereas if the value of the transform skip (enabled) flag is defined as 0, the syntax elements excluding the transform skip (enabled) flag syntax element may not be transmitted.

**[0144]** That is, a method for transmitting a high-level syntax element is proposed, which is dependent on whether to perform the transform skip, such as the minimum quantization parameter information for the transform skip block and whether to apply the BDPCM, during the transform skip in accordance with the value of the transform skip (enabled) flag in

the high-level syntax HLS (e.g., VPS, SPS, PPS, APS, DPS, and slice header). Further, the proposed method is not limited to the syntax elements mentioned in the present embodiment, but may include all high-level syntax elements defined in the high-level syntax set which has dependency depending on whether to perform the transform skip and which includes the transform skip (enabled) flag.

5 **[0145]** As described above, the syntax elements related to the transform skip based coding may be defined in the high-level syntax set, and may be defined in a sequence parameter set (SPS) as in an embodiment in Table 6 below.

[Table 6]

10

	Descriptor
seq_parameter_set_rbsp( ) {	
(...)	
<b>sps_transform_skip_enabled_flag</b>	u(1)
if( sps_transform_skip_enabled_flag) {	
<b>sps_bdpcm_enabled_flag</b>	u(1)
<b>min_qp_prime_ts_minus4</b>	ue(v)
}	
(...)	
rbsp_trailing_bits( )	
}	

15

20

25 **[0146]** Further, for example, the semantic of a syntax element for the above-described embodiment among syntax elements of the SPS syntax may be represented as in Table 7 below.

[Table 7]

30

**sps\_transform\_skip\_enabled\_flag** equal to 1 specifies that transform\_skip\_flag may be present in the transform unit syntax. sps\_transform\_skip\_enabled\_flag equal to 0 specifies that transform\_skip\_flag is not present in the transform unit syntax.

**sps\_bdpcm\_enabled\_flag** equal to 1 specifies that intra\_bdpcm\_flag may be present in the coding unit syntax for intra coding units. sps\_bdpcm\_enabled\_flag equal to 0 specifies that intra\_bdpcm\_flag is not present in the coding unit syntax for intra coding units. When not present, the value of sps\_bdpcm\_enabled\_flag is inferred to be equal to 0.

**min\_qp\_prime\_ts\_minus4** specifies the minimum allowed quantization parameter for transform skip mode as follows:

$$QpPrimeTsMin = 4 + \text{min\_qp\_prime\_ts\_minus4}$$

35

40

**[0147]** Referring to Table 6 and Table 7 above, syntax elements related to the transform skip may be defined in the SPS, and may include syntax elements of sps\_transform\_skip\_enabled\_flag, sps\_bdpcm\_enabled\_flag, and min\_qp\_prime\_ts\_minus4.

45

**[0148]** The sps\_transform\_skip\_enabled\_flag syntax element may represent whether the transform skip is enabled based on whether the value thereof is 0 or 1. For example, if the value of the sps\_transform\_skip\_enabled\_flag is 1, it may represent that the transform skip is enabled, and in this case, the transform\_skip\_flag may be parsed/signaled through a transform unit syntax. Here, the transform\_skip\_flag syntax element may represent whether the transform can be applied to the corresponding associated transform block. If the value of the sps\_transform\_skip\_enabled\_flag is 0, it may represent that the transform skip is not enabled, and in this case, the transform\_skip\_flag may not be parsed/signaled in the transform unit syntax. In other words, based on the transform skip enabled flag sps\_transform\_skip\_enabled\_flag, it may be represented whether the transform\_skip\_flag is present in the transform unit syntax.

50

55

**[0149]** The sps\_bdpcm\_enabled\_flag syntax element may represent whether the BDPCM is enabled based on whether the value thereof is 0 or 1. For example, if the value of sps\_bdpcm\_enabled\_flag is 1, it may represent that the BDPCM is enabled, and in this case, the intra\_bdpcm\_flag (or intra\_bdpcm\_luma\_flag and intra\_bdpcm\_chroma\_flag) may be parsed/signaled through the coding unit syntax for the intra coding unit. Here, the intra\_bdpcm\_flag syntax element may represent whether the BDPCM is applied to the current coding block. If the value of the sps\_bdpcm\_enabled\_flag is 0, it may represent that the BDPCM is not enabled, and in this case, the intra\_bdpcm\_flag (or intra\_bdpcm\_luma\_flag and intra\_bdpcm\_chroma\_flag) may not be parsed/signaled in the coding unit syntax for the intra coding unit. In other words, it may

be represented whether the `intra_bdpcm_flag` (or `intra_bdpcm_luma_flag` and `intra_bdpcm_chroma_flag`) is present in the coding unit syntax based on the BDPCM enabled flag `sps_bdpcm_enabled_flag`.

[0150] The `min_qp_prime_ts_minus4` syntax element may represent the minimum allowed quantization parameter allowed for the transform skip mode. For example, based on the `min_qp_prime_ts_minus4` syntax element, the minimum quantization parameter value (e.g., `QpPrimeTsMin`) in the transform skip mode may be derived. Based on the minimum quantization parameter in the transform skip mode, the quantization parameter being used in a scaling process (dequantization process) may be derived. Further, a scaled transform coefficient (dequantized transform coefficient) may be derived by performing the scaling process (dequantization process) for the current block based on the quantization parameter, and based on this, a residual sample of the current block may be derived.

[0151] Further, in the SPS, syntax elements having dependency may be defined with respect to the transform skip enabled flag syntax element (e.g., `sps_transform_skip_enabled_flag`) among the syntax elements related to the transform skip. For example, as disclosed in Table 6 and Table 7 above, based on the value of the transform skip enabled flag (e.g., `sps_transform_skip_enabled_flag`) in the SPS, the `min_qp_prime ts minus4` syntax element representing the minimum quantization parameter information for the transform skip block in the transform skip mode and the `sps_bdpcm_enabled_flag` syntax element representing whether the BDPCM is enabled may have dependency. As an example, if the value of the transform skip enabled flag (e.g., `sps_transform_skip_enabled_flag`) is 1, the `min_qp_prime_ts_minus4` syntax element and the `sps_bdpcm_enabled_flag` syntax element may be parsed/signaled. Further, if the value of the transform skip enabled flag (e.g., `sps_transform_skip_enabled_flag`) is 0, the `min_qp_prime_ts_minus4` syntax element and the `sps_bdpcm_enabled_flag` syntax element may not be parsed/signaled.

[0152] Further, as an embodiment, the present document proposes a method for saving bits being transmitted by enabling the syntax element having dependency to determine whether to perform parsing/signaling according to the dependent condition with respect to the transform skip related high-level syntax element. As an example, proposed is a method for parsing the high-level syntax element having dependency depending on whether the transform skip by the transform skip (enabled) flag is used.

[0153] For example, as syntax elements dependent on the transform skip based coding, there are a transform skip (enabled) flag (e.g., `sps_transform_skip_enabled_flag`), information on the transform skip application size (e.g., `log2_transform_skip_max_size_minus2`), minimum quantization parameter information for the transform skip (e.g., `min_qp_prime ts minus4`), and information on whether the BDPCM is applied (e.g., `sps_bdpcm_enabled_flag`). As an example, if the value of the transform skip (enabled) flag is defined as 1, the related flag or information syntax elements should be necessarily transmitted, whereas if the value of the transform skip (enabled) flag is defined as 0, the syntax elements excluding the transform skip (enabled) flag syntax element may not be transmitted.

[0154] That is, a method for transmitting a high-level syntax element is proposed, which is dependent on whether to perform the transform skip, such as information on the maximum size of the transform skip application, the minimum quantization parameter information, and whether to apply the BDPCM during the transform skip, in accordance with the value of the transform skip (enabled) flag in the high-level syntax HLS (e.g., VPS, SPS, PPS, APS, DPS, and slice header). Further, the proposed method is not limited to the syntax elements mentioned in the present embodiment, but may include all high-level syntax elements defined in the high-level syntax set which has dependency depending on whether to perform the transform skip and which includes the transform skip (enabled) flag.

[0155] As described above, the syntax elements related to the transform skip based coding may be defined in the high-level syntax set, and may be defined in the sequence parameter set (SPS) as in an embodiment of Table 8 below. However, the maximum block size information for the transform skip that is defined in a picture parameter set (PPS) in the related art may be newly defined in the SPS to avoid dependency between HLSs, and this may be represented as in Table 8 below.

[Table 8]

<code>seq_parameter_set_rbsp( ) {</code>	<b>Descriptor</b>
<code>(...)</code>	
<code>  sps_transform_skip_enabled_flag</code>	<code>u(1)</code>
<code>  if( sps_transform_skip_enabled_flag ) {</code>	
<code>    sps_bdpcm_enabled_flag</code>	<code>u(1)</code>
<code>    min_qp_prime_ts_minus4</code>	<code>ue(v)</code>
<code>    log2_transform_skip_max_size_minus2</code>	<code>ue(v)</code>
<code>  }</code>	
<code>(...)</code>	
<code>  rbsp_trailing_bits( )</code>	

(continued)

}	
---	--

5 **[0156]** Further, for example, the semantic of a syntax element for the above-described embodiment among syntax elements of the SPS syntax may be represented as in Table 9 below.

[Table 9]

10	<p><b>sps_transform_skip_enabled_flag</b> equal to 1 specifies that transform_skip_flag may be present in the transform unit syntax. sps_transform_skip_enabled_flag equal to 0 specifies that transform_skip_flag is not present in the transform unit syntax.</p>
15	<p><b>sps_bdpcm_enabled_flag</b> equal to 1 specifies that intra_bdpcm_flag may be present in the coding unit syntax for intra coding units. sps_bdpcm_enabled_flag equal to 0 specifies that intra_bdpcm_flag is not present in the coding unit syntax for intra coding units. When not present, the value of sps_bdpcm_enabled_flag is inferred to be equal to 0.</p>
20	<p><b>min_qp_prime_ts_minus4</b> specifies the minimum allowed quantization parameter for transform skip mode as follows:  <math display="block">QpPrimeTsMin = 4 + \text{min\_qp\_prime\_ts\_minus4}</math></p> <p><b>log2_transform_skip_max_size_minus2</b> specifies the maximum block size used for transform skip, and shall be in the range of 0 to 3.                  When not present, the value of log2_transform_skip_max_size_minus2 is inferred to be equal to 0.                  The variable MaxTsSize is set equal to <math>1 \ll (\text{log2\_transform\_skip\_max\_size\_minus2} + 2)</math>.</p>

25 **[0157]** Referring to Table 8 and Table 9 above, the syntax elements related to the transform skip may be defined in the SPS, and may include syntax elements of sps\_transform\_skip\_enabled\_flag, sps\_bdpcm\_enabled\_flag, min\_qp\_prime\_ts\_minus4, and log2\_transform\_skip\_max\_size\_minus2.

30 **[0158]** Here, since the syntax elements of the sps\_transform\_skip\_enabled\_flag, the sps\_bdpcm\_enabled\_flag, and the min\_qp\_prime\_ts\_minus4 have been explained in detail in Table 6 and Table 7 above, in the present embodiment, the detailed explanation thereof will be omitted for convenience in explanation.

35 **[0159]** The log2\_transform\_skip\_max\_size\_minus2 syntax element may represent the maximum block size being used in the transform skip mode. In this case, the log2\_transform\_skip\_max\_size\_minus2 syntax element may be in the range of 0 to 3. For example, as disclosed in Table 9 above, the maximum block size (e.g., MaxTsSize) being used in the transform skip mode may be derived based on calculation such as  $1 \ll (\text{log2\_transform\_skip\_max\_size\_minus2} + 2)$ .

40 **[0160]** Further, in the SPS, syntax elements having dependency may be defined with respect to the transform skip enabled flag syntax element (e.g., sps\_transform\_skip\_enabled\_flag) among syntax elements related to the transform skip. For example, as disclosed in Table 8 and Table 9 above, in the SPS, the sps\_bdpcm\_enabled\_flag syntax element representing whether the BDPCM is enabled based on the value of the transform skip enabled flag (e.g., sps\_transform\_skip\_enabled\_flag), the min\_qp\_prime\_ts\_minus4 syntax element representing the minimum quantization parameter information for the transform skip block in the transform skip mode, and the log2\_transform\_skip\_max\_size\_minus2 syntax element representing the maximum block size being used in the transform skip mode may have the dependency. As an example, if the value of the transform skip enabled flag (e.g., sps\_transform\_skip\_enabled\_flag) is 1, syntax elements of the sps\_bdpcm\_enabled\_flag, the min\_qp\_prime\_ts\_minus4, and the log2\_transform\_skip\_max\_size\_minus2 may be parsed/signaled. Further, if the value of the transform skip enabled flag (e.g., sps\_transform\_skip\_enabled\_flag) is 0, syntax elements of the sps\_bdpcm\_enabled\_flag, the min\_qp\_prime\_ts\_minus4, and the log2\_transform\_skip\_max\_size\_minus2 may not be parsed/signaled.

45 **[0161]** Further, as an embodiment, the present document proposes a method for saving bits being transmitted by enabling the syntax element having dependency to determine whether to perform parsing/signaling according to the dependent condition with respect to the transform skip related high-level syntax element and the palette coding related high-level syntax element. As an example, proposed is a method for parsing the high-level syntax element having dependency by the transform skip (enabled) flag and/or the palette coding (enabled) flag.

50 **[0162]** For example, as syntax elements dependent on the transform skip based coding, there are a transform skip (enabled) flag (e.g., sps\_transform\_skip\_enabled\_flag), information on the transform skip application size (e.g., log2\_transform\_skip\_max\_size\_minus2), minimum quantization parameter information for the transform skip (e.g., min\_qp\_prime\_ts\_minus4), and information on whether the BDPCM is applied (e.g., sps\_bdpcm\_enabled\_flag). Further, as described above, since the escape value is also not changed during the palette coding, the minimum quantization parameter information for the transform skip may be used in performing the quantization. Accordingly, the palette coding

(enabled) flag (e.g., `sps_palette_enabled_flag`) for the palette mode based coding and the minimum quantization parameter information (e.g., `min_qp_prime_ts_minus4`) during the transform skip may have dependency. As an example, if the value of the transform skip (enabled) flag or the palette coding (enabled) flag is defined as 1, the related flag or the information syntax elements should be necessarily transmitted, whereas if the value of the transform skip (enabled) flag or the palette coding (enabled) flag is defined as 0, the syntax elements excluding the respective flag syntax element may not be transmitted.

**[0163]** That is, a method for transmitting a high-level syntax element is proposed, which is dependent on whether to perform the transform skip, such as information on the maximum size of the transform skip application, the minimum quantization parameter information during the transform skip, and whether to apply the BDPCM, or whether to perform the palette coding in accordance with the value of the transform skip (enabled) flag and/or the value of the palette coding (enabled) flag in the high-level syntax (e.g., VPS, SPS, PPS, APS, DPS, and slice header).

**[0164]** For example, (i) In case that both the transform skip (enabled) flag and the palette coding (enabled) flag are defined as 1, syntax elements corresponding to a union of syntax elements dependent on the transform skip (enabled) flag and the palette coding (enabled) flag may be parsed. (ii) In case that the transform skip (enabled) flag is defined as 1, and the palette coding (enabled) flag is 0, the syntax elements dependent on the transform skip (enabled) flag may be parsed. (iii) In case that the transform skip (enabled) flag is defined as 0, and the palette coding (enabled) flag is 1, the syntax elements dependent on the palette coding (enabled) flag may be parsed. (iv) In case that both the transform skip (enabled) flag and the palette coding (enabled) flag are 0, other high-level syntax elements having dependency on two coding tools may not be parsed.

**[0165]** The parsing order of the syntax elements mentioned in the present embodiment is not specifically limited, and in case that whether to perform parsing is determined according to the dependency between the syntax elements, they are considered to coincide with each other. Further, the proposed method is not limited to the syntax elements mentioned in the present embodiment, but may have dependency depending on whether the transform skip or the palette coding is performed, and may include all high-level syntax elements defined in the high-level syntax set including the transform skip (enabled) flag and the palette coding (enabled) flag.

**[0166]** As described above, the syntax elements related to the transform skip based coding and/or the palette mode based coding may be defined in the high-level syntax set, and may be defined in the sequence parameter set (SPS) as in an embodiment of Table 10 below.

[Table 10]

	seq_parameter_set_rbsp() {	Descriptor
	(...)	
	<b>sps_transform_skip_enabled_flag</b>	u(1)
	if( <code>sps_transform_skip_enabled_flag</code> )	
	<b>sps_bdpcm_enabled_flag</b>	u(1)
	(...)	
	if( <code>chroma_format_idc == 3</code> )	
	<b>sps_palette_enabled_flag</b>	u(1)
	(...)	
	if( <code>sps_transform_skip_enabled_flag    sps_palette_enabled_flag</code> )	
	<b>min_qp_prime_ts_minus4</b>	ue(v)
	(...)	
	}	

**[0167]** Further, for example, the semantic of the syntax element for the above-described embodiment among the syntax elements of the SPS syntax may be represented as in Table 11 below.

[Table 11]

<p><b>sps_transform_skip_enabled_flag</b> equal to 1 specifies that <code>transform_skip_flag</code> may be present in the transform unit syntax. <code>sps_transform_skip_enabled_flag</code> equal to 0 specifies that <code>transform_skip_flag</code> is not present in the transform unit syntax.</p>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

(continued)

**sps\_bdpcm\_enabled\_flag** equal to 1 specifies that `intra_bdpcm_flag` may be present in the coding unit syntax for intra coding units. `sps_bdpcm_enabled_flag` equal to 0 specifies that `intra_bdpcm_flag` is not present in the coding unit syntax for intra coding units. When not present, the value of `sps_bdpcm_enabled_flag` is inferred to be equal to 0.

**min\_qp\_prime\_ts\_minus4** specifies the minimum allowed quantization parameter for transform skip mode as follows:

$$\text{QpPrimeTsMin} = 4 + \text{min\_qp\_prime\_ts\_minus4}$$

**sps\_palette\_enabled\_flag** equal to 1 specifies that `pred_mode_plt_flag` may be present in the coding unit syntax. `sps_palette_enabled_flag` equal to 0 specifies that `pred_mode_plt_flag` is not present in the coding unit syntax. When `sps_palette_enabled_flag` is not present, it is inferred to be equal to 0.

**[0168]** Referring to Table 10 and Table 11 above, in the SPS, syntax elements related to the transform skip and/or the palette coding may be defined, and may include syntax elements of the `sps_transform_skip_enabled_flag`, `sps_bdpcm_enabled_flag`, `sps_palette_enabled_flag`, and `min_qp_prime_ts_minus4`.

**[0169]** Here, since the syntax elements of the `sps_transform_skip_enabled_flag`, `sps_bdpcm_enabled_flag`, and `min_qp_prime_ts_minus4` have been explained in detail in Table 6 to Table 9 above, in the present embodiment, for convenience in explanation, the detailed explanation thereof will be omitted.

**[0170]** The `sps_palette_enabled_flag` syntax element may represent whether the palette coding (i.e., palette prediction mode) is enabled based on whether the value thereof is 0 or 1. For example, if the value of the `sps_palette_enabled_flag` is 1, it may represent that the palette coding is enabled, and in this case, the `pred_mode_plt_flag` may be parsed/signaled through the coding unit syntax. Here, the `pred_mode_plt_flag` syntax element may represent whether the palette mode can be used for the current coding unit. If the value of the `sps_palette_enabled_flag` is 0, it may represent that the palette coding is not enabled, and in this case, the `pred_mode_plt_flag` may not be parsed/signaled in the coding unit syntax. In other words, based on the palette coding enabled flag `sps_palette_enabled_flag`, it may represent whether the `pred_mode_plt_flag` is present in the coding unit syntax.

**[0171]** Further, in the SPS, syntax elements having dependency may be defined with respect to the transform skip enabled flag syntax element (e.g., `sps_transform_skip_enabled_flag`) among syntax elements related to the transform skip and/or the palette coding. For example, as disclosed in Table 10 and Table 11 above, in the SPS, the `sps_bdpcm_enabled_flag` syntax element representing whether the BDPCM is enabled based on the value of the transform skip enabled flag (e.g., `sps_transform_skip_enabled_flag`) may have dependency. As an example, if the value of the transform skip enabled flag (e.g., `sps_transform_skip_enabled_flag`) is 1, the `sps_bdpcm_enabled_flag` syntax element may be parsed/signaled. Further, if the value of the transform skip enabled flag (e.g., `sps_transform_skip_enabled_flag`) is 0, the `sps_bdpcm_enabled_flag` syntax element may not be parsed/signaled.

**[0172]** Further, in the SPS, a dependent condition may be defined in relation to the palette coding enabled flag syntax element (e.g., `sps_palette_enabled_flag`) among the syntax elements related to the transform skip and/or the palette coding. For example, as disclosed in Table 10 and Table 11 above, in the SPS, the palette coding enabled flag syntax element (e.g., `sps_palette_enabled_flag`) may be parsed/signaled based on the `chroma_format_idc` syntax element. As an example, if the value of the `chroma_format_idc` syntax element is 3, the `sps_palette_enabled_flag` syntax element may be parsed/signaled.

**[0173]** Further, in the SPS, syntax elements having dependency may be defined with respect to the transform skip enabled flag syntax element (e.g., `sps_transform_skip_enabled_flag`) and/or the palette coding enabled flag syntax element (e.g., `sps_palette_enabled_flag`) among the syntax elements related to the transform skip and/or the palette coding. For example, as disclosed in Table 10 and Table 11 above, in the SPS, the `min_qp_prime_ts_minus4` syntax element representing the minimum quantization parameter information for the transform skip mode may have dependency based on the value of the transform skip enabled flag (e.g., `sps_transform_skip_enabled_flag`) and/or the palette coding enabled flag syntax element (e.g., `sps_palette_enabled_flag`). As an example, in case that the value of the transform skip enabled flag (e.g., `sps_transform_skip_enabled_flag`) is 1, or the value of the palette coding enabled flag (e.g., `sps_palette_enabled_flag`) is 1, the `min_qp_prime_ts_minus4` syntax element may be parsed/signaled.

**[0174]** Further, as described above, the syntax elements related to the transform skip based coding and/or the palette mode based coding may be defined in the high-level syntax set, and as in an embodiment of Table 12 below, may be defined in the sequence parameter set (SPS). However, the maximum block size information for the transform skip that is defined in a picture parameter set (PPS) in the related art may be newly defined in the SPS to avoid dependency between HLSs, and this may be represented as in Table 12 below.

[Table 12]

		Descriptor
	seq_parameter_set_rbsp( ) {	
	(...)	
5	<b>sps_transform_skip_enabled_flag</b>	u(1)
	if( sps_transform_skip_enabled_flag ) {	
	<b>sps_bdpcm_enabled_flag</b>	u(1)
10	<b>log2_transform_skip_max_size_minus2</b>	ue(v)
	}	
	(...)	
	if( chroma_format_idc == 3 )	
15	<b>sps_palette_enabled_flag</b>	u(1)
	(...)	
	if( sps_transform_skip_enabled_flag    sps_palette_enabled_flag )	
20	<b>min_qp_prime_ts_minus4</b>	ue(v)
	(...)	
	}	

25 **[0175]** Further, for example, the semantic of a syntax element for the above-described embodiment among syntax elements of the SPS syntax may be represented as in Table 13 below.

[Table 13]

30	<b>sps_transform_skip_enabled_flag</b> equal to 1 specifies that transform_skip_flag may be present in the transform unit syntax. sps_transform_skip_enabled_flag equal to 0 specifies that transform_skip_flag is not present in the transform unit syntax.
35	<b>sps_bdpcm_enabled_flag</b> equal to 1 specifies that intra_bdpcm_flag may be present in the coding unit syntax for intra coding units. sps_bdpcm_enabled_flag equal to 0 specifies that intra_bdpcm_flag is not present in the coding unit syntax for intra coding units. When not present, the value of sps_bdpcm_enabled_flag is inferred to be equal to 0.
	<b>min_qp_prime_ts_minus4</b> specifies the minimum allowed quantization parameter for transform skip mode as follows: QpPrimeTsMin = 4 + min_qp_prime_ts_minus4
40	<b>log2_transform_skip_max_size_minus2</b> specifies the maximum block size used for transform skip, and shall be in the range of 0 to 3. When not present, the value of log2_transform_skip_max_size_minus2 is inferred to be equal to 0. The variable MaxTsSize is set equal to $1 \ll (\log_2 \text{transform\_skip\_max\_size\_minus2} + 2)$ .
45	<b>sps_palette_enabled_flag</b> equal to 1 specifies that pred_mode_plt_flag may be present in the coding unit syntax. sps_palette_enabled_flag equal to 0 specifies that pred_mode_plt_flag is not present in the coding unit syntax. When sps_palette_enabled_flag is not present, it is inferred to be equal to 0.

50 **[0176]** Referring to Table 12 and Table 13 above, in the SPS, syntax elements related to the transform skip and/or the palette coding may be defined, and may include syntax elements of the sps\_transform\_skip\_enabled\_flag, the sps\_bdpcm\_enabled\_flag, the log2\_transform\_skip\_max\_size\_minus2, the sps\_palette\_enabled\_flag, and the min\_qp\_prime\_ts\_minus4.

55 **[0177]** Here, since the syntax elements of the sps\_transform\_skip\_enabled\_flag, the sps\_bdpcm\_enabled\_flag, the log2\_transform\_skip\_max\_size\_minus2, the sps\_palette\_enabled\_flag, and the min\_qp\_prime\_ts\_minus4 have been described in detail in Table 6 to Table 11 above, in the present embodiment, for convenience in explanation, the detailed explanation thereof will be omitted.

**[0178]** As disclosed in an embodiment of Table 12 and Table 13 above, in the SPS, syntax elements having dependency may be defined with respect to the transform skip enabled flag syntax element (e.g., sps\_transform\_skip\_enabled\_flag)

among the syntax elements related to the transform skip and/or the palette coding. For example, as disclosed in Table 12 and Table 13 above, in the SPS, the `sps_bdpcm_enabled_flag` syntax element representing whether the BDPCM is enabled based on the value of the transform skip enabled flag (e.g., `sps_transform_skip_enabled_flag`), and the `log2_transform_skip_max_size_minus2` syntax element representing the maximum block size being used in the transform skip mode may have the dependency. As an example, if the value of the transform skip enabled flag (e.g., `sps_transform_skip_enabled_flag`) is 1, the syntax elements of the `sps_bdpcm_enabled_flag` and the `log2_transform_skip_max_size_minus2` may be parsed/signaled. Further, if the value of the transform skip enabled flag (e.g., `sps_transform_skip_enabled_flag`) is 0, the syntax elements of the `sps_bdpcm_enabled_flag` and the `log2_transform_skip_max_size_minus2` may not be parsed/signaled.

**[0179]** Further, in the SPS, syntax elements having dependency may be defined with respect to the transform skip enabled flag syntax element (e.g., `sps_transform_skip_enabled_flag`) and/or the palette coding enabled flag syntax element (e.g., `sps_palette_enabled_flag`) among the syntax elements related to the transform skip and/or the palette coding. For example, as disclosed in Table 12 and Table 13 above, the `min_qp_prime_ts_minus4` syntax element representing the minimum quantization parameter information for the transform skip mode based on the values of the transform skip enabled flag (e.g., `sps_transform_skip_enabled_flag`) and/or the palette coding enabled flag syntax element (e.g., `sps_palette_enabled_flag`) may have the dependency in the SPS. As an example, in case that the value of the transform skip enabled flag (e.g., `sps_transform_skip_enabled_flag`) is 1, or the value of the palette coding enabled flag (e.g., `sps_palette_enabled_flag`) is 1, the `min_qp_prime_ts_minus4` syntax element may be parsed/signaled.

**[0180]** Meanwhile, the source or the coded picture/image may include a luma component array, and in some cases, may further include two chroma components (cb, cr) array, That is, one pixel of the picture/image may include a luma sample and a chroma sample (cb, cr).

**[0181]** A color format may represent a configuration format of the luma component and the chroma component (cb, cr), and may be called a chroma format. The color format (or chroma format) may be predetermined, or may be adaptively signaled. For example, the chroma format may be signaled based on at least one of `chroma_format_idc` and `separate_colour_plane_flag` as in Table 14 below.

[Table 14]

<code>chroma_format_idc</code>	<code>separate_colour_plane_flag</code>	Chroma format	SubWidthC	SubHeightC
0	0	Monochrome	1	1
1	0	4:2:0	2	2
2	0	4:2:2	2	1
3	0	4:4:4	1	1
3	1	4:4:4	1	1

**[0182]** Referring to Table 14 above, in monochrome sampling, there is only one sample array that is nominally considered as a luma array.

**[0183]** In 4:2:0 sampling, each of two chroma arrays has a half height and a half width of the luma array.

**[0184]** In 4:2:2 sampling, each of two chroma arrays has the same height and a half width of the luma array.

**[0185]** In 4:4:4 sampling, the followings may be applied in accordance with the value of the `separate_colour_plane_flag`.

**[0186]** - If the value of the `separate_colour_plane_flag` is 0, each of the two chroma arrays has the same height and the same width of the luma array.

**[0187]** - Otherwise, if the value of the `separate_colour_plane_flag` is 1, three kinds of color planes may be separately processed as monochrome sampled pictures.

**[0188]** `SubWidthC` and `SubHeightC` may represent a ratio between the luma sample and the chroma sample. For example, if the `chroma_format_idc` is 3, the chroma format is 4:4:4, and in this case, in case that the width of the luma sample block is 16, the width of the corresponding chroma sample block may be  $16/\text{SubWidthC}$ . In general, the chroma sample related syntax and bitstream may be parsed only in case that the chroma array type (e.g., `chromaArrayType`) is not 0.

**[0189]** Further, as an embodiment, the present document proposes a method for saving bits being transmitted by enabling the syntax element having dependency to determine whether to perform parsing/signaling in accordance with the dependent condition with respect to the transform skip related high-level syntax element and the palette coding related high-level syntax element. As an example, a method for parsing the high-level syntax element having dependency by the transform skip (enabled) flag and the palette coding (enabled) flag is proposed.

**[0190]** For example, as syntax elements dependent on the transform skip based coding, there are a transform skip (enabled) flag (e.g., `sps_transform_skip_enabled_flag`), information on the size of the transform skip application (e.g.,

log2\_transform\_skip\_max\_size\_minus2), minimum quantization parameter information during the transform skip (e.g., min\_qp\_prime\_ts\_minus4), and information on whether to apply the BDPCM (e.g., sps\_bdpcm\_enabled\_flag). Further, as described above, since the escape value is also not changed during the palette coding, the minimum quantization parameter information for the transform skip may be used in performing the quantization.

**[0191]** As being represented in the above-described embodiment, if the value of the transform skip (enabled) flag or the palette coding (enabled) flag is defined as 1, the related flag or the information syntax elements should be necessarily transmitted, whereas if the value of the transform skip (enabled) flag or the palette coding (enabled) flag is defined as 0, the syntax elements excluding the respective flag syntax elements may not be transmitted. That is, a method for transmitting a high-level syntax element is proposed, which is dependent on whether to perform the transform skip, such as the minimum quantization parameter information during the transform skip or the palette coding and whether to apply the BDPCM, in accordance with the value of the transform skip (enabled) flag and the palette coding (enabled) flag, or whether to perform the palette coding in the high-level syntax (e.g., VPS, SPS, PPS, APS, DPS, and slice header).

**[0192]** For example, (i) In case that both the transform skip (enabled) flag and the palette coding (enabled) flag are defined as 1, syntax elements corresponding to a union of syntax elements dependent on the transform skip (enabled) flag and the palette coding (enabled) flag may be parsed. (ii) In case that the transform skip (enabled) flag is defined as 1, and the palette coding (enabled) flag is 0, the syntax elements dependent on the transform skip (enabled) flag may be parsed. (iii) In case that the transform skip (enabled) flag is defined as 0, and the palette coding (enabled) flag is 1, the syntax elements dependent on the palette coding (enabled) flag may be parsed. (iv) In case that both the transform skip (enabled) flag and the palette coding (enabled) flag are 0, other high-level syntax elements having dependency on two coding tools may not be parsed.

**[0193]** The parsing order of the syntax elements mentioned in the present embodiment is not specifically limited, and in case that whether to perform parsing is determined according to the dependency between the syntax elements, they are considered to coincide with each other. Further, the proposed method is not limited to the syntax elements mentioned in the present embodiment, but may have dependency depending on whether the transform skip or the palette coding is performed, and may include all high-level syntax elements defined in the high-level syntax set including the transform skip (enabled) flag and the palette coding (enabled) flag.

**[0194]** As described above, the syntax elements related to the transform skip based coding and/or the palette mode based coding may be defined in the high-level syntax set, and may be defined in the sequence parameter set (SPS) as in an embodiment of Table 15 below.

[Table 15]

seq_parameter_set_rbsp( ) {	Descriptor
(...)	
<b>sps_transform_skip_enabled_flag</b>	u(1)
if( sps_transform_skip_enabled_flag )	
<b>sps_bdpcm_enabled_flag</b>	u(1)
(...)	
if( chroma_format_idc == 3 )	
<b>sps_palette_enabled_flag</b>	u(1)
(...)	
if( sps_palette_enabled_flag )	
<b>min_qp_prime_ts_chroma_minus4</b>	ue(v)
(...)	
if( sps_transform_skip_enabled_flag    sps_palette_enabled_flag )	
<b>min_qp_prime_ts_luma_minus4</b>	ue(v)
(...)	
}	

**[0195]** Further, for example, the semantic of the syntax element for the above-described embodiment among the syntax elements of the SPS syntax may be represented as in Table 16 below.

[Table 16]

**sps\_transform\_skip\_enabled\_flag** equal to 1 specifies that transform\_skip\_flag may be present in the transform unit syntax. sps\_transform\_skip\_enabled\_flag equal to 0 specifies that transform\_skip\_flag is not present in the transform unit syntax.

**sps\_bdpcm\_enabled\_flag** equal to 1 specifies that intra\_bdpcm\_flag may be present in the coding unit syntax for intra coding units. sps\_bdpcm\_enabled\_flag equal to 0 specifies that intra\_bdpcm\_flag is not present in the coding unit syntax for intra coding units. When not present, the value of sps\_bdpcm\_enabled\_flag is inferred to be equal to 0.

**min\_qp\_prime\_ts\_luma\_minus4** specifies the minimum allowed quantization parameter for transform skip mode in the luma component as follows:

$$QpPrimeTsMin = 4 + \text{min\_qp\_prime\_ts\_luma\_minus4}$$

**min\_qp\_prime\_ts\_chroma\_minus4** specifies the minimum allowed quantization parameter for transform skip mode in the chroma component as follows:

$$QpPrimeTsMin = 4 + \text{min\_qp\_prime\_ts\_chroma\_minus4}$$

**sps\_palette\_enabled\_flag** equal to 1 specifies that pred\_mode\_plt\_flag may be present in the coding unit syntax. sps\_palette\_enabled\_flag equal to 0 specifies that pred\_mode\_plt\_flag is not present in the coding unit syntax. When sps\_palette\_enabled\_flag is not present, it is inferred to be equal to 0.

**[0196]** Referring to Table 15 and Table 16 above, in the SPS, syntax elements related to the transform skip and/or the palette coding may be defined, and may include syntax elements of the sps\_transform\_skip\_enabled\_flag, sps\_bdpcm\_enabled\_flag, sps\_palette\_enabled\_flag, min\_qp\_prime\_ts\_luma\_minus4, and min\_qp\_prime\_ts\_chroma\_minus4.

**[0197]** Here, since the syntax elements of the sps\_transform\_skip\_enabled\_flag, sps\_bdpcm\_enabled\_flag, and sps\_palette\_enabled\_flag have been explained in detail in Table 6 to Table 11 above, in the present embodiment, for convenience in explanation, the detailed explanation thereof will be omitted.

**[0198]** As disclosed in an embodiment of Table 15 and Table 16 above, in the SPS, syntax elements having dependency may be defined with respect to the transform skip enabled flag syntax element (e.g., sps\_transform\_skip\_enabled\_flag) among the syntax elements related to the transform skip and/or the palette coding. For example, as disclosed in Table 15 and Table 16 above, in the SPS, the sps\_bdpcm\_enabled\_flag syntax element representing whether the BDPCM is enabled based on the value of the transform skip enabled flag (e.g., sps\_transform\_skip\_enabled\_flag) may have the dependency. As an example, if the value of the transform skip enabled flag (e.g., sps\_transform\_skip\_enabled\_flag) is 1, the sps\_bdpcm\_enabled\_flag syntax element may be parsed/signaled. Further, if the value of the transform skip enabled flag (e.g., sps\_transform\_skip\_enabled\_flag) is 0, the sps\_bdpcm\_enabled\_flag syntax element may not be parsed/signaled.

**[0199]** Further, in the SPS, syntax elements having dependency may be defined with respect to the palette coding enabled flag syntax element (e.g., sps\_palette\_enabled\_flag) among the syntax elements related to the transform skip and/or the palette coding. For example, as disclosed in Table 15 and Table 16 above, the min\_qp\_prime\_ts\_chroma\_minus4 syntax element representing the minimum quantization parameter information in the transform skip mode for the chroma component based on the value of the palette coding enabled flag (e.g., sps\_palette\_enabled\_flag) may have the dependency in the SPS. As an example, if the value of the palette coding enabled flag (e.g., sps\_palette\_enabled\_flag) is 1, the min\_qp\_prime\_ts\_chroma\_minus4 syntax element may be parsed/signaled. Further, if the value of the palette coding enabled flag (e.g., sps\_palette\_enabled\_flag) is 0, the min\_qp\_prime\_ts\_chroma\_minus4 syntax element may not be parsed/signaled.

**[0200]** Further, in the SPS, syntax elements having dependency may be defined with respect to the transform skip enabled flag syntax element (e.g., sps\_transform\_skip\_enabled\_flag) and/or the palette coding enabled flag syntax element (e.g., sps\_palette\_enabled\_flag) among the syntax elements related to the transform skip and/or the palette coding. For example, as disclosed in Table 15 and Table 16 above, the min\_qp\_prime\_ts\_luma\_minus4 syntax element representing the minimum quantization parameter information in the transform skip mode for the luma component based on the value of the transform skip enabled flag (e.g., sps\_transform\_skip\_enabled\_flag) and/or the palette coding enabled flag syntax element (e.g., sps\_palette\_enabled\_flag) may have the dependency in the SPS. As an example, if the value of the transform skip enabled flag (e.g., sps\_transform\_skip\_enabled\_flag) is 1, or the value of the palette coding enabled flag (e.g., sps\_palette\_enabled\_flag) is 1, the min\_qp\_prime\_ts\_luma\_minus4 syntax element may be parsed/signaled.

**[0201]** Further, as an embodiment, the present document proposes a method for saving bits being transmitted by enabling the syntax element having dependency to determine whether to perform parsing/signaling in accordance with the dependent condition with respect to the transform skip related high-level syntax element and the palette coding related high-level syntax element. As an example, a method for parsing the high-level syntax element having dependency by the

transform skip (enabled) flag and the palette coding (enabled) flag is proposed.

[0202] For example, as syntax elements dependent on the transform skip based coding, there are a transform skip (enabled) flag (e.g., `sps_transform_skip_enabled_flag`), information on the size of the transform skip application (e.g., `log2_transform_skip_max_size_minus2`), minimum quantization parameter information during the transform skip (e.g., `min_qp_prime_ts_minus4`), and information on whether to apply the BDPCM (e.g., `sps_bdpcm_enabled_flag`). Further, as described above, since the escape value is also not changed during the palette coding, the minimum quantization parameter information for the transform skip may be used in performing the quantization.

[0203] As being represented in the above-described embodiment, if the value of the transform skip (enabled) flag or the palette coding (enabled) flag is defined as 1, the related flag or the information syntax elements should be necessarily transmitted, whereas if the value of the transform skip (enabled) flag or the palette coding (enabled) flag is defined as 0, the syntax elements excluding the respective flag syntax elements may not be transmitted. That is, a method for transmitting a high-level syntax element is proposed, which is dependent on whether to perform the transform skip, such as information on the maximum size of transform skip application, the minimum quantization parameter information during the transform skip or the palette coding, and whether to apply the BDPCM, in accordance with the value of the transform skip (enabled) flag and the palette coding (enabled) flag, or whether to perform the palette coding in the high-level syntax (e.g., VPS, SPS, PPS, APS, DPS, and slice header).

[0204] For example, (i) In case that both the transform skip (enabled) flag and the palette coding (enabled) flag are defined as 1, syntax elements corresponding to a union of syntax elements dependent on the transform skip (enabled) flag and the palette coding (enabled) flag may be parsed. (ii) In case that the transform skip (enabled) flag is defined as 1, and the palette coding (enabled) flag is 0, the syntax elements dependent on the transform skip (enabled) flag may be parsed. (iii) In case that the transform skip (enabled) flag is defined as 0, and the palette coding (enabled) flag is 1, the syntax elements dependent on the palette coding (enabled) flag may be parsed. (iv) In case that both the transform skip (enabled) flag and the palette coding (enabled) flag are 0, other high-level syntax elements having dependency on two coding tools may not be parsed.

[0205] The parsing order of the syntax elements mentioned in the present embodiment is not specifically limited, and in case that whether to perform parsing is determined according to the dependency between the syntax elements, they are considered to coincide with each other. Further, the proposed method is not limited to the syntax elements mentioned in the present embodiment, but may have dependency depending on whether the transform skip or the palette coding is performed, and may include all high-level syntax elements defined in the high-level syntax set including the transform skip (enabled) flag and the palette coding (enabled) flag.

[0206] As described above, the syntax elements related to the transform skip based coding and/or the palette mode based coding may be defined in the high-level syntax set, and may be defined in the sequence parameter set (SPS) as in an embodiment of Table 17 below. In the present embodiment, information on the maximum size of the transform skip defined in the picture parameter set (PPS) in the related art may be newly defined in the SPS to avoid the dependency between the HLSs, and a method for performing parsing/signaling based on the dependency of the transform skip and palette coding related syntax element being previously used is proposed.

[Table 17]

		Descriptor
40	<code>seq_parameter_set_rbsp( ) {</code>	
	<code>(...)</code>	
	<b><code>sps_transform_skip_enabled_flag</code></b>	<code>u(1)</code>
	<code>if( sps_transform_skip_enabled_flag ) {</code>	
45	<b><code>sps_bdpcm_enabled_flag</code></b>	<code>u(1)</code>
	<b><code>log2_transform_skip_max_size_minus2</code></b>	<code>ue(v)</code>
	<code>}</code>	
	<code>(...)</code>	
50	<code>if( chroma_format_idc == 3 )</code>	
	<b><code>sps_palette_enabled_flag</code></b>	<code>u(1)</code>
	<code>(...)</code>	
55	<code>if( sps_palette_enabled_flag )</code>	
	<b><code>min_qp_prime_ts_chroma_minus4</code></b>	<code>ue(v)</code>
	<code>(...)</code>	

(continued)

	if( sps_transform_skip_enabled_flag    sps_palette_enabled_flag )	
	min_qp_prime_ts_luma_minus4	ue(v)
5	(...)	
	}	

[0207] Further, for example, the semantic of the syntax element for the above-described embodiment among the syntax elements of the SPS syntax may be represented as in Table 18 below.

[Table 18]

15	<b>sps_transform_skip_enabled_flag</b> equal to 1 specifies that transform_skip_flag may be present in the transform unit syntax. sps_transform_skip_enabled_flag equal to 0 specifies that transform_skip_flag is not present in the transform unit syntax.
20	<b>sps_bdpcm_enabled_flag</b> equal to 1 specifies that intra_bdpcm_flag may be present in the coding unit syntax for intra coding units. sps_bdpcm_enabled_flag equal to 0 specifies that intra_bdpcm_flag is not present in the coding unit syntax for intra coding units. When not present, the value of sps_bdpcm_enabled_flag is inferred to be equal to 0.
25	<b>log2_transform_skip_max_size_minus2</b> specifies the maximum block size used for transform skip, and shall be in the range of 0 to 3. When not present, the value of log2_transform_skip_max_size_minus2 is inferred to be equal to 0. The variable MaxTsSize is set equal to $1 \ll (\log_2\_transform\_skip\_max\_size\_minus2 + 2)$ .
30	<b>min_qp_prime_ts_luma_minus4</b> specifies the minimum allowed quantization parameter for transform skip mode in the luma component as follows: QpPrimeTsMin = 4 + min_qp_prime_ts_luma_minus4
35	<b>min_qp_prime_ts_chroma_minus4</b> specifies the minimum allowed quantization parameter for transform skip mode in the chroma component as follows: QpPrimeTsMin = 4 + min_qp_prime_ts_chroma_minus4
	<b>sps_palette_enabled_flag</b> equal to 1 specifies that pred_mode_plt_flag may be present in the coding unit syntax. sps_palette_enabled_flag equal to 0 specifies that pred_mode_plt_flag is not present in the coding unit syntax. When sps_palette_enabled_flag is not present, it is inferred to be equal to 0.

[0208] Referring to Table 17 and Table 18 above, in the SPS, syntax elements related to the transform skip and/or the palette coding may be defined, and may include syntax elements of the sps\_transform\_skip\_enabled\_flag, sps\_bdpcm\_enabled\_flag, log2\_transform\_skip\_max\_size\_minus2, sps\_palette\_enabled\_flag, min\_qp\_prime\_ts\_luma\_minus4, and min\_qp\_prime\_ts\_chroma\_minus4.

[0209] Here, since the syntax elements of the sps\_transform\_skip\_enabled\_flag, sps\_bdpcm\_enabled\_flag, log2\_transform\_skip\_max\_size\_minus2, sps\_palette\_enabled\_flag, min\_qp\_prime\_ts\_luma\_minus4, and min\_qp\_prime\_ts\_chroma\_minus4 have been explained in detail in Table 6 to Table 11 above, in the present embodiment, for convenience in explanation, the detailed explanation thereof will be omitted.

[0210] As disclosed in an embodiment of Table 17 and Table 18 above, in the SPS, syntax elements having dependency may be defined with respect to the transform skip enabled flag syntax element (e.g., sps\_transform\_skip\_enabled\_flag) among the syntax elements related to the transform skip and/or the palette coding. For example, as disclosed in Table 17 and Table 18 above, in the SPS, the sps\_bdpcm\_enabled\_flag syntax element representing whether the BDPCM is enabled based on the value of the transform skip enabled flag (e.g., sps\_transform\_skip\_enabled\_flag) and the log2\_transform\_skip\_max\_size\_minus2 syntax element representing the maximum block size being used in the transform skip mode may have the dependency. As an example, if the value of the transform skip enabled flag (e.g., sps\_transform\_skip\_enabled\_flag) is 1, the syntax elements of the sps\_bdpcm\_enabled\_flag and the log2\_transform\_skip\_max\_size\_minus2 may be parsed/signaled. Further, if the value of the transform skip enabled flag (e.g., sps\_transform\_skip\_enabled\_flag) is 0, the syntax elements of the sps\_bdpcm\_enabled\_flag and the log2\_transform\_skip\_max\_size\_minus2 may not be parsed/signaled. Further, in the SPS, syntax elements having dependency may be defined with respect to the palette coding enabled flag syntax element (e.g., sps\_palette\_enabled\_flag) among the syntax elements related to the transform skip and/or the palette coding. For example, as disclosed in Table 17 and Table 18 above, the min\_qp\_prime\_ts\_chroma\_minus4 syntax element representing the minimum quantization parameter information in

the transform skip mode for the chroma component based on the value of the palette coding enabled flag (e.g., `sps_palette_enabled_flag`) may have the dependency in the SPS. As an example, if the value of the palette coding enabled flag (e.g., `sps_palette_enabled_flag`) is 1, the `min_qp_prime_ts_chroma_minus4` syntax element may be parsed/signaled. Further, if the value of the palette coding enabled flag (e.g., `sps_palette_enabled_flag`) is 0, the `min_qp_prime_ts_chroma_minus4` syntax element may not be parsed/signaled.

**[0211]** Further, in the SPS, syntax elements having dependency may be defined with respect to the transform skip enabled flag syntax element (e.g., `sps_transform_skip_enabled_flag`) and/or the palette coding enabled flag syntax element (e.g., `sps_palette_enabled_flag`) among the syntax elements related to the transform skip and/or the palette coding. For example, as disclosed in Table 17 and Table 18 above, the `min_qp_prime_ts_luma_minus4` syntax element representing the minimum quantization parameter information in the transform skip mode for the luma component based on the value of the transform skip enabled flag (e.g., `sps_transform_skip_enabled_flag`) and/or the palette coding enabled flag syntax element (e.g., `sps_palette_enabled_flag`) may have the dependency in the SPS. As an example, if the value of the transform skip enabled flag (e.g., `sps_transform_skip_enabled_flag`) is 1, or the value of the palette coding enabled flag (e.g., `sps_palette_enabled_flag`) is 1, the `min_qp_prime_ts_luma_minus4` syntax element may be parsed/signaled.

**[0212]** The following drawing has been prepared to explain a detailed example of the present document. Since the name of a detailed device or a detailed term or name (e.g., syntax/name of syntax element) described in the drawing is exemplarily presented, the technical features of the present document are not limited to the detailed name used in the drawing. FIGS. 10 and 11 schematically illustrate a video/image encoding method and an example of related components according to embodiment(s) of the present document.

**[0213]** FIGS. 10 and 11 schematically illustrate a video/image encoding method and an example of related components according to embodiment(s) of the present document.

**[0214]** The method disclosed in FIG. 10 may be performed by the encoding apparatus 200 disclosed in FIG. 2 or FIG. 11. Here, the encoding apparatus 200 disclosed in FIG. 11 briefly represents the encoding apparatus 200 disclosed in FIG. 2. Specifically, steps S1000 to S1010 of FIG. 10 may be performed by the residual processor 230 disclosed in FIG. 2, and step S1020 of FIG. 10 may be performed by the entropy encoder 240 disclosed in FIG. 2. Further, although not illustrated, a process of deriving a prediction sample may be performed by the predictor 220 of the encoding apparatus 200, a process of generating a reconstructed sample for the current block and a reconstructed picture based on the residual sample and the prediction sample for the current block may be performed by the adder 250 of the encoding apparatus 200, and a process of encoding prediction information for the current block may be performed by the entropy encoder 240 of the encoding apparatus 200. Further, the method disclosed in FIG. 10 may include the above-described embodiments of the present document to be performed. Accordingly, referring to FIG. 10, the detailed explanation of the duplicate contents of the above-described embodiments will be omitted or simplified.

**[0215]** Referring to FIG. 10, the encoding apparatus determines whether to apply transform to the current block based on transform skip enabled information (S1000).

**[0216]** As an embodiment, the encoding apparatus may first determine the prediction mode for the current block, and may derive prediction samples. For example, the encoding apparatus may determine whether to perform inter prediction or intra prediction with respect to the current block, and may also determine a specific inter prediction mode or a specific intra prediction mode based on an RD cost. Further, the encoding apparatus may determine whether to perform the prediction with respect to the current block based on a CIIP mode, an IBC mode, a BDPCM mode, or a palette mode. The encoding apparatus may derive the prediction samples for the current block by performing the prediction in accordance with the determined prediction mode. In this case, various prediction methods disclosed in the present document, such as inter prediction or intra prediction, may be applied. Further, the encoding apparatus may generate and encode information (e.g., prediction mode information) related to the prediction applied to the current block.

**[0217]** Further, the encoding apparatus may derive the residual samples by comparing the prediction samples with the original samples for the current block with each other. The encoding apparatus may derive transform coefficients through a transform process for the residual samples. In this case, the encoding apparatus may determine whether to apply the transform to the current block in consideration of the coding efficiency. That is, the encoding apparatus may determine whether the transform is applied to the residual samples of the current block.

**[0218]** For example, the encoding apparatus determines whether to apply the transform or the transform skip mode to the current block (residual samples) based on transform skip enabled information.

**[0219]** As described above, the transform skip enabled information is information on whether the transform skip is enabled, and as disclosed in Table 6 to Table 18, it may be represented as the `sps_transform_skip_enabled_flag` syntax element. For example, if the value of the `sps_transform_skip_enabled_flag` is 1, it represents that the transform skip is enabled, and in this case, the `transform_skip_flag` is parsed/signaled through the transform unit syntax. Here, the `transform_skip_flag` syntax element represents whether the transform can be applied to the associated transform block. If the value of the `sps_transform_skip_enabled_flag` is 0, it represents that the transform skip is not enabled, and in this case, the `transform_skip_flag` is not parsed/signaled in the transform unit syntax. The transform skip enabled information (e.g., `sps_transform_skip_enabled_flag`) may be included in the SPS, and may be signaled to the decoding apparatus. That is,

based on that the value of the transform skip enabled information (e.g., `sps_transform_skip_enabled_flag`) included in the SPS is 1, the transform unit syntax may include the transform skip flag (e.g., `transform_skip_flag`). In this case, if the value of the transform skip flag (e.g., `transform_skip_flag`) included in the transform unit syntax is 1, a mode in which the transform is not applied (transform skip mode) may be performed for the current block. Further, if the value of the transform skip flag (e.g., `transform_skip_flag`) included in the transform unit syntax is 0, the transform is applied to the current block.

**[0220]** For example, if the value of the transform skip enabled information is 1 (i.e., with respect to the transform skip enabled information representing that the transform skip is enabled), the encoding apparatus determines whether to apply the transform to the current block. That is, the encoding apparatus generates information (transform skip flag) on whether to apply the transform to the current block based on that the value of the transform skip enabled information is 1, and signals the transform skip flag through the transform unit syntax. In this case, if the transform is not applied to the current block (i.e., in case of the transform skip mode), the encoding apparatus generates a transform skip flag of which the value is 1, and includes this in the transform unit syntax. Further, in case of applying the transform to the current block, the encoding apparatus generates the transform skip flag of which the value is 0, and includes this in the transform unit syntax.

**[0221]** The encoding apparatus generates residual information for the current block based on whether to apply the transform (S 1010).

**[0222]** As an embodiment, the encoding apparatus derives residual samples of the current block, and generates residual information by applying the transform or transform skip to the residual samples of the current block based on whether to apply the transform. For example, with respect to the residual samples for the current block of which the transform skip flag value is 1, the encoding apparatus applies the transform skip mode. In this case, the encoding apparatus derives the residual samples of the current block as transform coefficients. Further, with respect to the residual samples for the current block of which the transform skip flag value is 0, the encoding apparatus derives the transform coefficients by performing the transform. The encoding apparatus may derive quantized transform coefficients by performing a quantization process based on the transform coefficients derived through the transform skip or the transform. The encoding apparatus may generate the residual information based on the quantized transform coefficients.

**[0223]** Here, the residual information may be information generated through the transform and/or quantization process, and may be information about the quantized transform coefficients, and for example, may include information on value information of the quantized transform coefficients, position information, transform technique, transform kernel, and quantization parameter.

**[0224]** The encoding apparatus encodes image information (or video information) (S1020).

**[0225]** Here, the image information may include the residual information. Further, the image information may include information (e.g., prediction mode information) related to the prediction used to derive the prediction samples. Further, the image information may include information related to the transform skip, for example, transform skip enabled information and transform skip flag information. That is, the image information may include various kinds of information derived in the encoding process, and the image information including such various kinds of information may be encoded.

**[0226]** Further, the image information may include various kinds of information according to the above-described embodiment(s) in the present document, and may include information disclosed in at least one of Tables 1 to 18 as described above.

**[0227]** For example, the image information may include a sequence parameter set (SPS). The SPS may include transform skip related information and palette coding related information. As an example, the transform skip related information may include transform skip enabled information (e.g., `sps_transform_skip_enabled_flag`), BDPCM enabled information (e.g., `sps_bdpcm_enabled_flag`), information on the maximum block size used in the transform skip mode (e.g., `log2_transform_skip_max_size_minus2`), and the minimum quantization parameter information related to the minimum allowed quantization parameter for the transform skip mode (e.g., `min_qp_prime_ts_minus4`). Further, as an example, the palette coding related information may include palette coding enabled information (e.g., `sps_palette_enabled_flag`) and the minimum quantization parameter information related to the minimum allowed quantization parameter for the transform skip mode (e.g., `min_qp_prime_ts_minus4`).

**[0228]** Further, for example, as described above, among the information related to the transform skip and/or the palette coding included in the SPS, information having dependency may be defined with respect to the transform skip enabled information (e.g., `sps_transform_skip_enabled_flag`).

**[0229]** As an example, the SPS may be configured to parse/signal BDPCM enabled flag information (e.g., `sps_bdpcm_enabled_flag`) on whether to enable the BDPCM based on the value of the transform skip enabled information (e.g., `sps_transform_skip_enabled_flag`). In this case, if the value of the transform skip enabled flag information (e.g., `sps_transform_skip_enabled_flag`) is 1, the BDPCM enabled flag information (e.g., `sps_bdpcm_enabled_flag`) may be included in the SPS, and the information (e.g., `sps_bdpcm_enabled_flag`) may be parsed/signaled from the SPS. Further, if the value of the transform skip enabled flag information (e.g., `sps_transform_skip_enabled_flag`) is 0, the BDPCM enabled flag information (e.g., `sps_bdpcm_enabled_flag`) may not be parsed/signaled from the SPS.

**[0230]** Further, based on the value of the BDPCM enabled flag information (e.g., `sps_bdpcm_enabled_flag`) in the SPS, the BDPCM flag information (e.g., `intra_bdpcm_flag`) on whether to apply the BDPCM to the current block) may be

parsed/signaled through the coding unit syntax. In this case, if the value of the BDPCM enabled flag information (e.g., `sps_bdpcm_enabled_flag`) is 1, the BDPCM flag information (e.g., `intra_bdpcm_flag`) may be included in the coding unit syntax, and the information (e.g., `intra_bdpcm_flag`) may be parsed/signaled from the coding unit syntax. Further, if the value of the BDPCM enabled flag information (e.g., `sps_bdpcm_enabled_flag`) is 0, the BDPCM flag information (e.g., `intra_bdpcm_flag`) may not be parsed/signaled from the coding unit syntax.

**[0231]** Further, as an example, the SPS may be configured to parse/signal information on the maximum block size (e.g., `log2_transform_skip_max_size_minus2`) used in the transform skip mode based on the value of the transform skip enabled information (e.g., `sps_transform_skip_enabled_flag`). In this case, if the value of the transform skip enabled flag information (e.g., `sps_transform_skip_enabled_flag`) is 1, the information on the maximum block size (e.g., `log2_transform_skip_max_size_minus2`) used in the transform skip mode may be included in the SPS, and the information (e.g., `log2_transform_skip_max_size_minus2`) may be parsed/signaled from the SPS. Further, if the value of the transform skip enabled flag information (e.g., `sps_transform_skip_enabled_flag`) is 0, the information on the maximum block size (e.g., `log2_transform_skip_max_size_minus2`) used in the transform skip mode may not be parsed/signaled from the SPS.

**[0232]** Further, as an example, based on the value of the transform skip enabled flag information (e.g., `sps_transform_skip_enabled_flag`) defined in the SPS, transform skip flag information (e.g., `transform_skip_flag`) on whether to apply the transform skip to the current block may be parsed/signaled through the transform unit syntax. In this case, if the value of the transform skip enabled flag information (e.g., `sps_transform_skip_enabled_flag`) is 1, the transform skip flag information (e.g., `transform_skip_flag`) may be included in the transform unit syntax, and the information (e.g., `transform_skip_flag`) may be parsed/signaled from the transform unit syntax. Further, if the value of the transform skip enabled flag information (e.g., `sps_transform_skip_enabled_flag`) is 0, the transform skip flag information (e.g., `transform_skip_flag`) may not be parsed/signaled from the transform unit syntax.

**[0233]** Further, for example, among the information related to the transform skip and/or the palette coding included in the SPS as described above, information having dependency may be defined with respect to the palette coding enabled information on whether the palette coding is enabled (e.g., `sps_palette_enabled_flag`). As an example, based on the value of the palette coding enabled information (e.g., `sps_palette_enabled_flag`) defined in the SPS, palette prediction mode flag information (e.g., `pred_mode_plt_flag`) on whether to apply the palette coding (palette prediction mode) to the current block may be parsed/signaled through the coding unit syntax. In this case, if the value of the palette coding enabled information (e.g., `sps_palette_enabled_flag`) is 1, the palette prediction mode flag information (e.g., `pred_mode_plt_flag`) may be included in the coding unit syntax, and the information (e.g., `pred_mode_plt_flag`) may be parsed/signaled from the coding unit syntax. Further, if the value of the palette coding enabled information (e.g., `sps_palette_enabled_flag`) is 0, the palette prediction mode flag information (e.g., `pred_mode_plt_flag`) may not be parsed/signaled from the coding unit syntax.

**[0234]** Further, for example, as described above, among the information related to the transform skip and/or the palette coding included in the SPS, information having dependency may be defined with respect to the transform skip enabled information (e.g., `sps_transform_skip_enabled_flag`) and/or the palette coding enabled information (e.g., `sps_palette_enabled_flag`).

**[0235]** As an example, in the SPS, based on at least one of the transform skip enabled information (e.g., `sps_transform_skip_enabled_flag`) and/or the palette coding enabled information (e.g., `sps_palette_enabled_flag`), the minimum quantization parameter information (e.g., `min_qp_prime_ts_minus4`) related to the minimum allowed quantization parameter for the transform skip mode is parsed/signaled. In other words, based on the condition that the value of the transform skip enabled information (e.g., `sps_transform_skip_enabled_flag`) is 1, or the value of the palette coding enabled information (e.g., `sps_palette_enabled_flag`) is 1, the minimum quantization parameter information (e.g., `min_qp_prime_ts_minus4`) may be included in the SPS, and only in case that the above condition is satisfied, the minimum quantization parameter information (e.g., `min_qp_prime_ts_minus4`) is parsed/signaled.

**[0236]** Here, as described above, the minimum quantization parameter information (e.g., `min_qp_prime_ts_minus4`) is information related to the minimum allowed quantization parameter for the transform skip mode, and based on this, the quantization parameter for the current block is derived.

**[0237]** For example, in case that the transform skip mode is applied to the current block, the quantization parameter for the current block is derived based on the minimum quantization parameter information (e.g., `min_qp_prime_ts_minus4`), and the quantized transform coefficients may be derived by performing the quantization process based on the quantization parameter.

**[0238]** Further, for example, in case that the palette coding mode is applied to the current block, the quantization parameter for an escape value of the current block is derived based on the minimum quantization parameter information (e.g., `min_qp_prime_ts_minus4`). In this case, the quantized escape value (e.g., `palette_escape_val`) may be derived by applying the quantization parameter to the escape value of the current block based on the quantization parameter. The process in which the palette coding mode is applied may be performed as disclosed in Table 4 and Table 5 above.

**[0239]** As described above, the image information including various kinds of information may be encoded and output in

the form of a bitstream. The bitstream may be transmitted to the decoding apparatus through a network or a (digital) storage medium. Here, 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, and SSD.

**[0240]** FIGS. 12 and 13 schematically illustrate a video/image decoding method and an example of related components according to embodiment(s) of the present document.

**[0241]** The method disclosed in FIG. 12 may be performed by the decoding apparatus 300 disclosed in FIG. 3 or FIG. 13. Here, the decoding apparatus 300 disclosed in FIG. 13 briefly represents the decoding apparatus 300 disclosed in FIG. 3. Specifically, step S1200 of FIG. 12 may be performed by the entropy decoder 310 disclosed in FIG. 3, steps S1210 to S1220 of FIG. 12 may be performed by the residual processor 320 disclosed in FIG. 3, and step S1230 of FIG. 12 may be performed by the adder 340 disclosed in FIG. 3. Further, although not illustrated, a process of receiving prediction information for the current block may be performed by the entropy decoder 310 of the decoding apparatus 300, and a process of deriving a prediction sample of the current block may be performed by the predictor 330 of the decoding apparatus 300. Further, the method disclosed in FIG. 12 may include the above-described embodiments of the present document to be performed. Accordingly, referring to FIG. 12, the detailed explanation of the duplicate contents of the above-described embodiments will be omitted or simplified.

**[0242]** Referring to FIG. 12, the decoding apparatus receives image information (or video information) from a bitstream (S 1200).

**[0243]** As an embodiment, the decoding apparatus derives information (e.g., video/image information) necessary for image reconstruction (or picture reconstruction) by parsing the bitstream. In this case, the image information includes residual information, and the residual information may include value information of quantized transform coefficients, location information, transform technique, transform kernel, and quantization parameter information. Further, the image information may include prediction-related information (e.g., prediction mode information). Further, the image information includes information related to the transform skip, for example, transform skip enabled information and transform skip flag information. That is, the image information may include various kinds of information necessary in the decoding process, and may be decoded based on a coding method, such as exponential Golomb coding, CAVLC, or CABAC.

**[0244]** Further, the image information may include various kinds of information according to the above-described embodiments of the present document, and may include information disclosed in at least one of Tables 1 to 18 as described above.

**[0245]** For example, the image information may include a sequence parameter set (SPS). The SPS may include transform skip related information and palette coding related information. As an example, the transform skip related information may include transform skip enabled information (e.g., `sps_transform_skip_enabled_flag`), BDPCM enabled information (e.g., `sps_bdpcm_enabled_flag`), information on the maximum block size used in the transform skip mode (e.g., `log2_transform_skip_max_size_minus2`), and the minimum quantization parameter information related to the minimum allowed quantization parameter for the transform skip mode (e.g., `min_qp_prime_ts_minus4`). Further, as an example, the palette coding related information may include palette coding enabled information (e.g., `sps_palette_enabled_flag`) and the minimum quantization parameter information related to the minimum allowed quantization parameter for the transform skip mode (e.g., `min_qp_prime_ts_minus4`).

**[0246]** Further, for example, as described above, among the information related to the transform skip and/or the palette coding included in the SPS, information having dependency may be defined with respect to the transform skip enabled information (e.g., `sps_transform_skip_enabled_flag`).

**[0247]** As an example, the SPS may be configured to parse/signal BDPCM enabled flag information (e.g., `sps_bdpcm_enabled_flag`) on whether to enable the BDPCM based on the value of the transform skip enabled information (e.g., `sps_transform_skip_enabled_flag`). In this case, if the value of the transform skip enabled flag information (e.g., `sps_transform_skip_enabled_flag`) is 1, the BDPCM enabled flag information (e.g., `sps_bdpcm_enabled_flag`) may be included in the SPS, and the information (e.g., `sps_bdpcm_enabled_flag`) may be parsed/signaled from the SPS. Further, if the value of the transform skip enabled flag information (e.g., `sps_transform_skip_enabled_flag`) is 0, the BDPCM enabled flag information (e.g., `sps_bdpcm_enabled_flag`) may not be parsed/signaled from the SPS.

**[0248]** Further, based on the value of the BDPCM enabled flag information (e.g., `sps_bdpcm_enabled_flag`) in the SPS, the BDPCM flag information (e.g., `intra_bdpcm_flag`) on whether to apply the BDPCM to the current block) may be parsed/signaled through the coding unit syntax. In this case, if the value of the BDPCM enabled flag information (e.g., `sps_bdpcm_enabled_flag`) is 1, the BDPCM flag information (e.g., `intra_bdpcm_flag`) may be included in the coding unit syntax, and the information (e.g., `intra_bdpcm_flag`) may be parsed/signaled from the coding unit syntax. Further, if the value of the BDPCM enabled flag information (e.g., `sps_bdpcm_enabled_flag`) is 0, the BDPCM flag information (e.g., `intra_bdpcm_flag`) may not be parsed/signaled from the coding unit syntax.

**[0249]** Further, as an example, the SPS may be configured to parse/signal information on the maximum block size (e.g., `log2_transform_skip_max_size_minus2`) used in the transform skip mode based on the value of the transform skip enabled information (e.g., `sps_transform_skip_enabled_flag`). In this case, if the value of the transform skip enabled flag information (e.g., `sps_transform_skip_enabled_flag`) is 1, the information on the maximum block size (e.g., `log2_trans-`

form\_skip\_max\_size\_minus2) used in the transform skip mode may be included in the SPS, and the information (e.g., log2\_transform\_skip\_max\_size\_minus2) may be parsed/signaled from the SPS. Further, if the value of the transform skip enabled flag information (e.g., sps\_transform\_skip\_enabled\_flag) is 0, the information on the maximum block size (e.g., log2\_transform\_skip\_max\_size\_minus2) used in the transform skip mode may not be parsed/signaled from the SPS.

5 **[0250]** Further, as an example, based on the value of the transform skip enabled flag information (e.g., sps\_transform\_skip\_enabled\_flag) defined in the SPS, transform skip flag information (e.g., transform\_skip\_flag) on whether to apply the transform skip to the current block may be parsed/signaled through the transform unit syntax. In this case, if the value of the transform skip enabled flag information (e.g., sps\_transform\_skip\_enabled\_flag) is 1, the transform skip flag information (e.g., transform\_skip\_flag) may be included in the transform unit syntax, and the information (e.g., transform\_skip\_flag) may be parsed/signaled from the transform unit syntax. Further, if the value of the transform skip enabled flag information (e.g., sps\_transform\_skip\_enabled\_flag) is 0, the transform skip flag information (e.g., transform\_skip\_flag) may not be parsed/signaled from the transform unit syntax.

10 **[0251]** Further, for example, among the information related to the transform skip and/or the palette coding included in the SPS as described above, information having dependency may be defined with respect to the palette coding enabled information on whether the palette coding is enabled (e.g., sps\_palette\_enabled\_flag). As an example, based on the value of the palette coding enabled information (e.g., sps\_palette\_enabled\_flag) defined in the SPS, palette prediction mode flag information (e.g., pred\_mode\_plt\_flag) on whether to apply the palette coding (palette prediction mode) to the current block may be parsed/signaled through the coding unit syntax. In this case, if the value of the palette coding enabled information (e.g., sps\_palette\_enabled\_flag) is 1, the palette prediction mode flag information (e.g., pred\_mode\_plt\_flag) may be included in the coding unit syntax, and the information (e.g., pred\_mode\_plt\_flag) may be parsed/signaled from the coding unit syntax. Further, if the value of the palette coding enabled information (e.g., sps\_palette\_enabled\_flag) is 0, the palette prediction mode flag information (e.g., pred\_mode\_plt\_flag) may not be parsed/signaled from the coding unit syntax.

15 **[0252]** Further, for example, as described above, among the information related to the transform skip and/or the palette coding included in the SPS, information having dependency may be defined with respect to the transform skip enabled information (e.g., sps\_transform\_skip\_enabled\_flag) and/or the palette coding enabled information (e.g., sps\_palette\_enabled\_flag).

20 **[0253]** As an example, in the SPS, based on at least one of the transform skip enabled information (e.g., sps\_transform\_skip\_enabled\_flag) and/or the palette coding enabled information (e.g., sps\_palette\_enabled\_flag), the minimum quantization parameter information (e.g., min\_qp\_prime\_ts\_minus4) related to the minimum allowed quantization parameter for the transform skip mode may be parsed/signaled. In other words, based on the condition that the value of the transform skip enabled information (e.g., sps\_transform\_skip\_enabled\_flag) is 1, or the value of the palette coding enabled information (e.g., sps\_palette\_enabled\_flag) is 1, the minimum quantization parameter information (e.g., min\_qp\_prime\_ts\_minus4) may be included in the SPS, and only in case that the above condition is satisfied, the minimum quantization parameter information (e.g., min\_qp\_prime\_ts\_minus4) may be parsed/signaled.

25 **[0254]** Here, as described above, the minimum quantization parameter information (e.g., min\_qp\_prime\_ts\_minus4) may be information related to the minimum allowed quantization parameter for the transform skip mode, and based on this, the quantization parameter for the current block may be derived.

30 **[0255]** For example, in case that the transform skip mode is applied to the current block, the quantization parameter for the current block may be derived based on the minimum quantization parameter information (e.g., min\_qp\_prime\_ts\_minus4), and the dequantized transform coefficient (scaled transform coefficient) may be derived by performing a dequantization process (scaling process) based on the quantization parameter. Based on the dequantized transform coefficient, the residual sample of the current block may be derived.

35 **[0256]** Further, for example, in case that the palette coding mode is applied to the current block, the quantization parameter for an escape value of the current block may be derived based on the minimum quantization parameter information (e.g., min\_qp\_prime\_ts\_minus4). In this case, the escape value of the current block may be derived by performing the dequantization (scaling process) based on the quantization parameter. Based on the escape value, the reconstructed sample of the current block may be generated. The process in which the palette coding mode is applied may be performed as disclosed in Table 4 and Table 5 above.

40 **[0257]** The decoding apparatus determines whether to apply the transform to the current block based on the transform skip enabled information (S1210).

45 **[0258]** As an embodiment, if the image information including the transform skip enabled information is received, the decoding apparatus determines whether to apply the transform or the transform skip mode to the current block based on the transform skip enabled information.

50 **[0259]** As described above, the transform skip enabled information is information on whether the transform skip is enabled, and may be represented as the sps\_transform\_skip\_enabled\_flag syntax element as disclosed in Table 6 to Table 18. For example, if the value of the sps\_transform\_skip\_enabled\_flag is 1, it may represent that the transform skip is enabled, and in this case, the transform\_skip\_flag may be parsed/signaled through the transform unit syntax. Here, the

transform\_skip\_flag syntax element may represent whether the transform can be applied to the associated transform block. If the value of the sps\_transform\_skip\_enabled\_flag is 0, it may represent that the transform skip is not enabled, and in this case, the transform\_skip\_flag may not be parsed/signaled in the transform unit syntax. The transform skip enabled information (e.g., sps\_transform\_skip\_enabled\_flag) may be included in the SPS, and may be signaled from the encoding apparatus to the decoding apparatus. That is, based on that the value of the transform skip enabled information (e.g., sps\_transform\_skip\_enabled\_flag) included in the SPS is 1, the transform unit syntax may include the transform skip flag (e.g., transform\_skip\_flag). In this case, if the value of the transform skip flag (e.g., transform\_skip\_flag) included in the transform unit syntax is 1, a mode in which the transform is not applied (transform skip mode) may be performed for the current block. Further, if the value of the transform skip flag (e.g., transform\_skip\_flag) included in the transform unit syntax is 0, the transform may be applied for the current block.

**[0260]** For example, if the value of the transform skip enabled information is 1 (i.e., for the transform skip enabled information representing that the transform skip is enabled), the decoding apparatus determines whether to apply the transform to the current block.

**[0261]** The decoding apparatus derives the residual sample based on whether to apply the transform and the residual information (S 1220).

**[0262]** As an embodiment, the decoding apparatus receives the image information including the residual information. As described above, the residual information may include value information of the quantized transform coefficients, position information, transform technique, transform kernel, and quantization parameter information. The decoding apparatus may derive the quantized transform coefficients for the current block based on the quantized transform coefficient information included in the residual information, and may derive the transform coefficients based on the quantized transform coefficients. Further, the decoding apparatus may derive the residual samples based on the transform coefficients.

**[0263]** For example, if the value of the transform skip enabled information is 1 (i.e., for the transform skip enabled information representing that the transform skip is enabled), the decoding apparatus obtains the information on whether to apply the transform to the current block (transform skip flag) from the transform unit syntax. In this case, the decoding apparatus derives the residual samples based on the transform skip flag information. For example, the transform skip mode is applied to the current block of which the value of the transform skip flag is 1, and in this case, the decoding apparatus derives the transform coefficients as the residual samples of the current block. Further, the transform is applied to the current block of which the value of the transform skip flag is 0, and in this case, the decoding apparatus derives the residual samples of the current block through inverse transform of the transform coefficients.

**[0264]** Further, with respect to the current block of which the value of the transform skip flag is 1 (i.e., transform skip mode), the decoding apparatus may derive the quantization parameter being used in the dequantization process based on the minimum quantization parameter information. Further, the decoding apparatus may derive the dequantized transform coefficients by performing the dequantization process based on the quantization parameter, and may derive the residual samples based on the dequantized transform coefficients.

**[0265]** Here, as described above, the minimum quantization parameter information related to the minimum allowed quantization parameter for the transform skip mode, and may be included in the image information (e.g., SPS) based on at least one of the transform skip enabled information (e.g., sps\_transform\_skip\_enabled\_flag) and/or the palette coding enabled information (e.g., sps\_palette\_enabled\_flag). For example, based on the condition that the value of the transform skip enabled information (e.g., sps\_transform\_skip\_enabled\_flag) is 1, or the value of the palette coding enabled information (e.g., sps\_palette\_enabled\_flag) is 1, the minimum quantization parameter information (e.g., min\_qp\_prime\_ts\_minus4) is included in the SPS. That is, only in case that the above condition is satisfied, the minimum quantization parameter information (e.g., min\_qp\_prime\_ts\_minus4) is parsed/signaled.

**[0266]** The decoding apparatus generates the reconstructed samples based on the residual samples (S1230).

**[0267]** As an embodiment, the decoding apparatus may determine whether to perform inter prediction or intra prediction for the current block based on the prediction information (e.g., prediction mode information) included in the image information, and may derive the prediction samples for the current block by performing the prediction in accordance with the determination. Further, the decoding apparatus may generate the reconstructed samples based on the prediction samples and the residual samples. In this case, the decoding apparatus may directly use the prediction samples as the reconstructed samples in accordance with the prediction mode, or may generate the reconstructed samples by adding the residual samples to the prediction samples. Further, the decoding apparatus may derive a reconstructed block or a reconstructed picture based on the reconstructed samples. Thereafter, as needed, the decoding apparatus may apply the in-loop filtering procedure, such as the deblocking filtering and/or SAO procedure, to the reconstructed picture in order to improve the subjective/objective image quality as described above.

**[0268]** In the above-described embodiments, the methods are explained on the basis of flowcharts by means of a series of steps or blocks, but the present document is not limited to the order of steps, and a certain step may be performed in order or step different from that described above, or concurrently with other steps. Further, it may be understood by a person having ordinary skill in the art that the steps shown in a flowchart are not exclusive, and that another step may be incorporated or one or more steps of the flowchart may be removed without affecting the scope of the present document.

**[0269]** The method according to the present document may be implemented in the form of software, and the encoding apparatus and/or decoding apparatus according to the present document may be included in an apparatus that performs image processing, such as TV, a computer, a smartphone, a set-top box, and a display apparatus.

**[0270]** When the embodiments of the present document are implemented by software, the aforementioned method may be implemented by a module (process or function) which performs the aforementioned function. The module may be stored in a memory and executed by a processor. The memory may be installed inside or outside the processor and may be connected to the processor via various well-known means. The processor may include Application-Specific Integrated Circuit (ASIC), other chipsets, a logical circuit, and/or a data processing device. The memory may include a Read-Only Memory (ROM), a Random Access Memory (RAM), a flash memory, a memory card, a storage medium, and/or other storage device. In other words, the embodiments according to the present document may be implemented and executed on a processor, a micro-processor, a controller, or a chip. For example, functional units illustrated in the respective figures may be implemented and executed on a computer, a processor, a microprocessor, a controller, or a chip. In this case, information on implementation (for example, information on instructions) or algorithms may be stored in a digital storage medium.

**[0271]** In addition, the decoding apparatus and the encoding apparatus to which the present document is applied may be included in a multimedia broadcasting transceiver, a mobile communication terminal, a home cinema video device, a digital cinema video device, a surveillance camera, a video chat device, and a real time communication device such as video communication, a mobile streaming device, a storage medium, a camcorder, a video on demand (VoD) service provider, an Over The Top (OTT) video device, an internet streaming service provider, a 3D video device, a Virtual Reality (VR) device, an Augment Reality (AR) device, an image telephone video device, a vehicle terminal (for example, a vehicle (including an autonomous vehicle) terminal, an airplane terminal, or a ship terminal), and a medical video device; and may be used to process an image signal or data. For example, the OTT video device may include a game console, a Blu-ray player, an Internet-connected TV, a home theater system, a smartphone, a tablet PC, and a Digital Video Recorder (DVR).

**[0272]** In addition, the processing method to which the present document is applied may be produced in the form of a program executed by a computer and may be stored in a computer-readable recording medium. Multimedia data having a data structure according to the present document may also be stored in the computer-readable recording medium. The computer readable recording medium includes all kinds of storage devices and distributed storage devices in which computer readable data is stored. The computer-readable recording medium may include, for example, a Blu-ray disc (BD), a universal serial bus (USB), a ROM, a PROM, an EPROM, an EEPROM, a RAM, a CD-ROM, a magnetic tape, a floppy disk, and an optical data storage device. The computer-readable recording medium also includes media embodied in the form of a carrier wave (for example, transmission over the Internet). In addition, a bitstream generated by the encoding method may be stored in the computer-readable recording medium or transmitted through a wired or wireless communication network.

**[0273]** In addition, the embodiment(s) of the present document may be embodied as a computer program product based on a program code, and the program code may be executed on a computer according to the embodiment(s) of the present document. The program code may be stored on a computer-readable carrier.

**[0274]** FIG. 14 represents an example of a contents streaming system to which the embodiment of the present document may be applied.

**[0275]** Referring to FIG. 14, the content streaming system to which the embodiments of the present document is applied may generally include an encoding server, a streaming server, a web server, a media storage, a user device, and a multimedia input device.

**[0276]** The encoding server functions to compress to digital data the contents input from the multimedia input devices, such as the smart phone, the camera, the camcorder and the like, to generate a bitstream, and to transmit it to the streaming server. As another example, in a case where the multimedia input device, such as, the smart phone, the camera, the camcorder or the like, directly generates a bitstream, the encoding server may be omitted.

**[0277]** The bitstream may be generated by an encoding method or a bitstream generation method to which the embodiments of the present document is applied. And the streaming server may temporarily store the bitstream in a process of transmitting or receiving the bitstream.

**[0278]** The streaming server transmits multimedia data to the user equipment on the basis of a user's request through the web server, which functions as an instrument that informs a user of what service there is. When the user requests a service which the user wants, the web server transfers the request to the streaming server, and the streaming server transmits multimedia data to the user. In this regard, the contents streaming system may include a separate control server, and in this case, the control server functions to control commands/responses between respective equipment in the content streaming system.

**[0279]** The streaming server may receive contents from the media storage and/or the encoding server. For example, in a case the contents are received from the encoding server, the contents may be received in real time. In this case, the streaming server may store the bitstream for a predetermined period of time to provide the streaming service smoothly.

**[0280]** For example, the user equipment may include a mobile phone, a smart phone, a laptop computer, a digital

broadcasting terminal, a personal digital assistant (PDA), a portable multimedia player (PMP), a navigation, a slate PC, a tablet PC, an ultrabook, a wearable device (e.g., a watch-type terminal (smart watch), a glass-type terminal (smart glass), a head mounted display (HMD)), a digital TV, a desktop computer, a digital signage or the like.

[0281] Each of servers in the contents streaming system may be operated as a distributed server, and in this case, data received by each server may be processed in distributed manner.

## Claims

- 10 1. An image decoding method for Versatile Video Coding, VVC, performed by a decoding apparatus, the method comprising:

obtaining image information including residual information and transform skip enabled information from a bitstream;

- 15 obtaining a transform skip flag information based on a value of the transform skip enabled information, wherein the transform skip enabled information is related to whether the transform skip is enabled, and i) the value of the transform skip enabled information equal to 1 represents that the transform skip is enabled and the transform skip flag information is present in a transform unit syntax, and ii) the value of the transform skip enabled information equal to 0 represents that the transform skip is not enabled and the transform skip flag information is not present in the transform unit syntax;

determining whether a transform is applied to a current block based on the transform skip flag information;

deriving a residual sample based on whether the transform is applied and the residual information; and

generating a reconstructed sample based on the residual sample,

- 25 wherein the image information includes palette coding enabled information for representing whether a palette mode is enabled, and wherein i) a value of the palette coding enabled information equal to 1 represents that the palette mode is enabled, and ii) the value of the palette coding enabled information equal to 0 represents that the palette mode is not enabled,

30 wherein based on a condition that the value of the transform skip enabled information is equal to 1 and a condition that the value of the palette coding enabled information is equal to 1, the image information further includes minimum quantization parameter information related to a minimum allowed quantization parameter for a transform skip mode, and wherein based on a condition that both the value of the transform skip enabled information and the value of the palette coding enabled information are equal to 0, the minimum quantization parameter information is not included in the image information,

- 35 wherein based on a case where the transform skip is applied to the current block, a quantization parameter for the current block is derived based on the minimum quantization parameter information, and wherein based on a case where the palette mode is applied to the current block, the quantization parameter for the current block is derived based on the minimum quantization parameter information,

40 wherein based on the case where the transform skip is applied to the current block, the residual sample is derived based on the quantization parameter, and based on the case where the palette mode is applied to the current block, an escape value is derived based on the quantization parameter.

2. An image encoding method for Versatile Video Coding, VVC, performed by an encoding apparatus, the method comprising:

45 determining whether to apply a transform to a current block based on transform skip enabled information;

generating a transform skip flag information based on whether the transform skip is applied;

generating residual information on the current block based on whether the transform is applied; and

encoding image information including the transform skip enabled information and the residual information,

- 50 wherein the transform skip enabled information is related to whether the transform skip is enabled, and i) a value of the transform skip enabled information equal to 1 represents that the transform skip is enabled and the transform skip flag information is present in a transform unit syntax, and ii) the value of the transform skip enabled information equal to 0 represents that the transform skip is not enabled and the transform skip flag information is not present in the transform unit syntax,

55 wherein the image information includes palette coding enabled information for representing whether a palette mode is enabled, and wherein i) a value of the palette coding enabled information equal to 1 represents that the palette mode is enabled, and ii) the value of the palette coding enabled information equal to 0 represents that the palette mode is not enabled,

wherein based on a condition that the value of the transform skip enabled information is equal to 1 and a condition

that the value of the palette coding enabled information is equal to 1, the image information further includes minimum quantization parameter information related to a minimum allowed quantization parameter for a transform skip mode, and wherein based on a condition that both the value of the transform skip enabled information and the value of the palette coding enabled information are equal to 0, the minimum quantization parameter information is not included in the image information,

wherein based on a case where the transform skip is applied to the current block, a quantization parameter for the current block is determined based on the minimum quantization parameter information, and wherein based on a case where the palette mode is applied to the current block, the quantization parameter for the current block is determined based on the minimum quantization parameter information.

3. A non-transitory computer readable storage medium storing a bitstream for Versatile Video Coding, VVC, generated by a method, the method comprising:

performing determining whether a transform is applied to a current block based on transform skip enabled information, generating a transform skip flag information based on whether the transform is applied, generating residual information on the current block based on whether the transform skip is applied, and generating the bitstream by encoding image information including the transform skip enabled information and the residual information,

wherein the transform skip enabled information is related to whether the transform skip is enabled, and i) a value of the transform skip enabled information equal to 1 represents that the transform skip is enabled and the transform skip flag information is present in a transform unit syntax, and ii) the value of the transform skip enabled information equal to 0 represents that the transform skip is not enabled and the transform skip flag information is not present in the transform unit syntax,

wherein the image information includes palette coding enabled information for representing whether a palette mode is enabled, and wherein i) a value of the palette coding enabled information equal to 1 represents that the palette mode is enabled, and ii) the value of the palette coding enabled information equal to 0 represents that the palette mode is not enabled,

wherein based on a condition that the value of the transform skip enabled information is equal to 1 and a condition that the value of the palette coding enabled information is equal to 1, the image information further includes minimum quantization parameter information related to a minimum allowed quantization parameter for a transform skip mode, and wherein based on a condition that both the value of the transform skip enabled information and the value of the palette coding enabled information are equal to 0, the minimum quantization parameter information is not included in the image information,

wherein based on a case where the transform skip is applied to the current block, a quantization parameter for the current block is determined based on the minimum quantization parameter information, and wherein based on a case where the palette mode is applied to the current block, the quantization parameter for the current block is determined based on the minimum quantization parameter information.

4. A transmission method of data for image, for Versatile Video Coding, VVC, the transmission method comprising:

generating a bitstream by performing the image encoding method of claim 2; and transmitting the data comprising the bitstream.

**Patentansprüche**

1. Bildecodierungsverfahren für Versatile Video Coding, VVC, durchgeführt von einer Decodierungsvorrichtung, wobei das Verfahren umfasst:

Erhalten von Bildinformation einschließlich Restinformation und Transformationsüberspringungs-Freigabeinformation aus einem Bitstrom;

Erhalten einer Transformationsüberspringungs-Flag-Information basierend auf einem Wert der Transformationsüberspringungs-Freigabeinformation, wobei die Transformationsüberspringungs-Freigabeinformation sich darauf bezieht, ob die Transformationsüberspringung freigegeben ist, und i) der Wert der Transformationsüberspringungs-Freigabeinformation gleich 1 anzeigt, dass die Transformationsüberspringung freigegeben ist und die Transformationsüberspringungs-Flag-Information in einer Transformationseinheitssyntax vorhanden ist, und ii) der Wert der Transformationsüberspringungs-Freigabeinformation gleich 0 anzeigt, dass die Transformationsüberspringung nicht freigegeben ist und die Transformationsüberspringungs-Flag-Information in der Transformationseinheitssyntax nicht vorhanden ist;

Bestimmen, ob eine Transformation auf einen aktuellen Block angewandt wird, basierend auf der Transformationsüberspringungs-Flag-Information;  
 Ableiten eines Restabstastwerts auf der Grundlage der Tatsache, ob die Transformation angewendet wird, und der Restinformation; und

Erzeugen eines rekonstruierten Abstastwerts auf der Grundlage des Restabstastwerts, wobei die Bildinformation Palettencodierungsfreigabeinformation enthält zur Darstellung, ob ein Palettenmodus aktiviert ist, und wobei i) ein Wert der Palettencodierungsfreigabeinformation gleich 1 bedeutet, dass der Palettenmodus aktiviert ist, und ii) der Wert der Palettencodierungsfreigabeinformation gleich 0 bedeutet, dass der Palettenmodus nicht aktiviert ist,

wobei basierend auf einer Bedingung, dass der Wert der Transformationsüberspringungsfreigabeinformation gleich 1 ist, und einer Bedingung, dass der Wert der Palettencodierungsfreigabeinformation gleich 1 ist, die Bildinformation ferner eine minimale Quantisierungsparameterinformation enthält, die sich auf einen minimal zulässigen Quantisierungsparameter für einen Transformationsüberspringungsmodus bezieht, und wobei basierend auf einer Bedingung, dass sowohl der Wert der Transformationsüberspringungsfreigabeinformation als auch der Wert der Palettencodierungsfreigabeinformation gleich 0 ist, die Mindestquantisierungsparameterinformation nicht in der Bildinformation enthalten ist,

wobei basierend auf einem Fall, in dem der Transformationsübersprung auf den aktuellen Block angewendet wird, ein Quantisierungsparameter für den aktuellen Block basierend auf der minimalen Quantisierungsparameterinformation abgeleitet wird, und wobei basierend auf einem Fall, in dem der Palettenmodus auf den aktuellen Block angewendet wird, der Quantisierungsparameter für den aktuellen Block basierend auf der minimalen Quantisierungsparameterinformation abgeleitet wird,

wobei basierend auf dem Fall, in dem der Transformationsübersprung auf den aktuellen Block angewendet wird, der Restabstastwert basierend auf dem Quantisierungsparameter abgeleitet wird, und basierend auf dem Fall, in dem der Palettenmodus auf den aktuellen Block angewendet wird, ein Escape-Wert basierend auf dem Quantisierungsparameter abgeleitet wird.

2. Ein Bildcodierungsverfahren für Versatile Video Coding, VVC, das von einer Codierungsvorrichtung durchgeführt wird, wobei das Verfahren umfasst:

Bestimmen, ob eine Transformation auf einen aktuellen Block angewendet werden soll, basierend auf einer Transformationsüberspringungsfreigabeinformation;  
 Erzeugen einer Transformationsüberspringungs-Flag-Information basierend darauf, ob die Transformationsüberspringung angewendet wird;

Erzeugen von Restinformation über den aktuellen Block, je nachdem, ob die Transformation angewendet wird; und

Kodieren von Bildinformation einschließlich der Transformationsüberspringungsfreigabeinformation und der Restinformation,

wobei die Transformationsüberspringungsfreigabeinformation sich darauf bezieht, ob die Transformationsüberspringung aktiviert ist, und i) ein Wert der Transformationsüberspringungsfreigabeinformation gleich 1 bedeutet, dass die Transformationsüberspringung aktiviert ist und die Transformationsüberspringungs-Flag-Information in einer Transformationseinheitssyntax vorhanden ist, und ii) der Wert der Transformationsüberspringungsfreigabeinformation gleich 0 bedeutet, dass die Transformationsüberspringung nicht aktiviert ist und die Transformationsüberspringungs-Flag-Information nicht in der Transformationseinheitssyntax vorhanden ist,

wobei die Bildinformation Palettencodierungsfreigabeinformation enthält zur Darstellung, ob ein Palettenmodus aktiviert ist, und wobei i) ein Wert der Palettencodierungsfreigabeinformation gleich 1 bedeutet, dass der Palettenmodus aktiviert ist, und ii) der Wert der Palettencodierungsfreigabeinformation gleich 0 bedeutet, dass der Palettenmodus nicht aktiviert ist,

wobei basierend auf einer Bedingung, dass der Wert der Transformationsüberspringungsfreigabeinformation gleich 1 ist, und einer Bedingung, dass der Wert der Palettencodierungsfreigabeinformation gleich 1 ist, die Bildinformation ferner eine minimale Quantisierungsparameterinformation enthält, die sich auf einen minimal zulässigen Quantisierungsparameter für einen Transformationsüberspringungsmodus bezieht, und wobei basierend auf einer Bedingung, dass sowohl der Wert der Transformationsüberspringungsfreigabeinformation als auch der Wert der Palettencodierungsinformation gleich 0 ist, die minimale Quantisierungsparameterinformation nicht in der Bildinformation enthalten ist,

wobei basierend auf einem Fall, in dem die Transformationsüberspringung auf den aktuellen Block angewendet wird, ein Quantisierungsparameter für den aktuellen Block basierend auf der minimalen Quantisierungsparameterinformation bestimmt wird, und wobei basierend auf einem Fall, in dem der Palettenmodus auf den aktuellen Block angewendet wird, der Quantisierungsparameter für den aktuellen Block basierend auf der minimalen

Quantisierungsparameterinformation bestimmt wird.

3. Nicht-transitorisches computerlesbares Speichermedium, das einen Bitstrom für Versatile Video Coding, VVC, speichert, der durch ein Verfahren erzeugt wird, wobei das Verfahren umfasst: Durchführen eines Bestimmens, ob eine Transformation auf einen aktuellen Block angewendet wird, auf der Grundlage von Transformationsüberspringungsfreigabeinformation ermöglichen, Erzeugen einer Transformationsüberspringungs-Flag-Information auf der Grundlage, ob die Transformation angewendet wird, Erzeugen von Restinformationen über den aktuellen Block auf der Grundlage, ob die Transformationsüberspringung angewendet wird, und Erzeugen des Bitstroms durch Codieren von Bildinformation, einschließlich der Transformationsüberspringungsfreigabeinformation und der Restinformation,

wobei die Transformationsüberspringungsfreigabeinformation sich darauf bezieht, ob die Transformationsüberspringung aktiviert ist, und i) ein Wert der Transformationsüberspringungsfreigabeinformation gleich 1 bedeutet, dass die Transformationsüberspringung aktiviert ist und die Transformationsüberspringungs-Flag-Information in einer Transformationseinheitssyntax vorhanden ist, und ii) der Wert der Transformationsüberspringungsfreigabeinformation gleich 0 bedeutet, dass die Transformationsüberspringung nicht aktiviert ist und die Transformationsüberspringungs-Flag-Information nicht in der Transformationseinheitssyntax vorhanden ist, wobei die Bildinformation Palettencodierungsfreigabeinformation enthält, um darzustellen, ob ein Palettenmodus aktiviert ist, und wobei i) ein Wert der Palettencodierungsfreigabeinformation gleich 1 bedeutet, dass der Palettenmodus aktiviert ist, und ii) der Wert der Palettencodierungsfreigabeinformation gleich 0 bedeutet, dass der Palettenmodus nicht aktiviert ist, wobei basierend auf einer Bedingung, dass der Wert der Transformationsüberspringungsfreigabeinformation gleich 1 ist, und einer Bedingung, dass der Wert der Palettencodierungsinformation gleich 1 ist, die Bildinformation ferner eine minimale Quantisierungsparameterinformation enthält, die sich auf einen minimal zulässigen Quantisierungsparameter für einen Transformationsüberspringungsmodus bezieht, und wobei basierend auf einer Bedingung, dass sowohl der Wert der Transformationsüberspringungsfreigabeinformation als auch der Wert der Palettencodierungsfreigabeinformation gleich 0 ist, die minimale Quantisierungsparameterinformation nicht in der Bildinformation enthalten ist, wobei basierend auf einem Fall, in dem die Transformationsüberspringung auf den aktuellen Block angewendet wird, ein Quantisierungsparameter für den aktuellen Block basierend auf der minimalen Quantisierungsparameterinformation bestimmt wird, und wobei basierend auf einem Fall, in dem der Palettenmodus auf den aktuellen Block angewendet wird, der Quantisierungsparameter für den aktuellen Block basierend auf der minimalen Quantisierungsparameterinformation bestimmt wird.

4. Übertragungsverfahren von Bilddaten für Versatile Video Coding, VVC, wobei das Übertragungsverfahren umfasst:

Erzeugen eines Bitstroms durch Ausführen des Bildcodierungsverfahrens nach Anspruch 2; und Übermitteln der Daten, die den Bitstrom umfassen.

## Revendications

1. Procédé de décodage d'image pour le codage vidéo polyvalent, VVC, exécuté par un appareil de décodage, le procédé comprenant:

l'obtention d'informations d'image comprenant des informations résiduelles et des informations de saut de transformation activé à partir d'un flux binaire;

l'obtention d'informations d'indicateur de saut de transformation en fonction de la valeur des informations de saut de transformation activé, les informations de saut de transformation activé indiquant si le saut de transformation est activé, et i) la valeur des informations de saut de transformation activé égale à 1 indique que le saut de transformation est activé et que les informations d'indication de saut de transformation sont présentes dans la syntaxe de l'unité de transformation, et ii) la valeur des informations de saut de transformation activé égale à 0 indique que le saut de transformation n'est pas activé et que les informations d'indication de saut de transformation ne sont pas présentes dans la syntaxe de l'unité de transformation;

la détermination si une transformation est appliquée à un bloc actuel sur la base des informations d'indicateur de saut de transformation;

la déduction d'un échantillon résiduel en fonction de l'application ou non de la transformation et des informations résiduelles; et

la génération d'un échantillon reconstruit sur la base de l'échantillon résiduel,  
 dans lequel les informations d'image comprennent des informations de codage de palette activé pour indiquer si  
 un mode palette est activé, et dans lequel i) une valeur des informations de codage de palette activé égale à 1  
 indique que le mode palette est activé, et ii) la valeur des informations de codage de palette activé égale à 0  
 indique que le mode palette n'est pas activé,  
 dans lequel, à condition que la valeur des informations de saut de transformation activé soit égale à 1 et que la  
 valeur des informations de codage de palette activé soit égale à 1, les informations d'image comprennent en outre  
 des informations de paramètre de quantification minimum concernant un paramètre de quantification minimum  
 autorisé pour un mode de saut de transformation, et dans lequel, à condition que la valeur des informations de  
 saut de transformation activé et la valeur des informations de codage de palette soient toutes deux égales à 0, les  
 informations de paramètre de quantification minimum ne sont pas incluses dans les informations d'image,  
 dans lequel, dans le cas où le saut de transformation serait appliqué au bloc actuel, un paramètre de  
 quantification pour le bloc actuel est dérivé sur la base des informations de paramètre de quantification minimum,  
 et dans lequel, dans le cas où le mode palette serait appliqué au bloc actuel, le paramètre de quantification pour le  
 bloc actuel est dérivé sur la base des informations de paramètre de quantification minimum,  
 dans lequel, dans le cas où le saut de transformation serait appliqué au bloc actuel, l'échantillon résiduel est  
 dérivé sur la base du paramètre de quantification, et dans le cas où le mode palette serait appliqué au bloc actuel,  
 une valeur d'échappement est dérivée sur la base du paramètre de quantification.

2. Procédé de codage d'image pour le codage vidéo polyvalent, VVC, exécuté par un appareil de codage, le procédé comprenant:

la détermination s'il faut appliquer une transformation à un bloc actuel sur la base des informations de saut de transformation activé;

la génération d'informations d'indicateur de saut de transformation en fonction de l'application ou non du saut de transformation;

la génération d'informations résiduelles sur le bloc actuel en fonction de l'application ou non de la transformation;  
 et

le codage d'informations d'image comprenant les informations de saut de transformation activé et des informations résiduelles,

dans lequel les informations de saut de transformation activé indiquant si le saut de transformation est activé, et i)  
 une valeur des informations de saut de transformation activé égale à 1 indique que le saut de transformation est  
 activé et que les informations d'indication de saut de transformation sont présentes dans la syntaxe de l'unité de  
 transformation, et ii) la valeur des informations de saut de transformation activé égale à 0 indique que le saut de  
 transformation n'est pas activé et que les informations d'indication de saut de transformation ne sont pas  
 présentes dans la syntaxe de l'unité de transformation,

dans lequel les informations d'image comprennent des informations de codage de palette activé pour indiquer si  
 un mode palette est activé, et dans lequel i) une valeur des informations de codage de palette activé égale à 1  
 indique que le mode palette est activé, et ii) la valeur des informations de codage de palette activé égale à 0  
 indique que le mode palette n'est pas activé,

dans lequel, à condition que la valeur des informations de saut de transformation activé soit égale à 1 et que la  
 valeur des informations de codage de palette activé soit égale à 1, les informations d'image comprennent en outre  
 des informations de paramètre de quantification minimum concernant un paramètre de quantification minimum  
 autorisé pour un mode de saut de transformation, et dans lequel, à condition que la valeur des informations de  
 saut de transformation activé et la valeur des informations de codage de palette soient toutes deux égales à 0, les  
 informations de paramètre de quantification minimum ne sont pas incluses dans les informations d'image,  
 dans lequel, dans le cas où le saut de transformation serait appliqué au bloc actuel, un paramètre de  
 quantification pour le bloc actuel est déterminé sur la base des informations de paramètre de quantification  
 minimum, et dans lequel, dans le cas où le mode palette serait appliqué au bloc actuel, le paramètre de  
 quantification pour le bloc actuel est déterminé sur la base des informations de paramètre de quantification  
 minimum.

3. Support d'enregistrement non transitoire lisible par ordinateur stockant un flux binaire pour le codage vidéo polyvalent, VVC, généré selon un procédé, le procédé comprenant: la détermination de l'application d'une transformation à un bloc actuel sur la base d'informations de saut de transformation activé, la génération d'informations d'indication de saut de transformation sur la base de l'application de la transformation, la génération d'informations résiduelles sur le bloc actuel sur la base de l'application du saut de transformation, et la génération du flux binaire par le codage d'informations d'image comprenant les informations de saut de transformation activé et les informations

résiduelles,

5 dans lequel les informations de saut de transformation activé indiquant si le saut de transformation est activé, et i) une valeur des informations de saut de transformation activé égale à 1 indique que le saut de transformation est activé et que les informations d'indication de saut de transformation sont présentes dans la syntaxe de l'unité de transformation, et ii) la valeur des informations de saut de transformation activé égale à 0 indique que le saut de transformation n'est pas activé et que les informations d'indication de saut de transformation ne sont pas présentes dans la syntaxe de l'unité de transformation,

10 dans lequel les informations d'image comprennent des informations de codage de palette activé pour indiquer si un mode palette est activé, et dans lequel i) une valeur des informations de codage de palette activé égale à 1 indique que le mode palette est activé, et ii) la valeur des informations de codage de palette activé égale à 0 indique que le mode palette n'est pas activé,

15 dans lequel, à condition que la valeur des informations de saut de transformation activé soit égale à 1 et que la valeur des informations de codage de palette activé soit égale à 1, les informations d'image comprennent en outre des informations de paramètre de quantification minimum concernant un paramètre de quantification minimum autorisé pour un mode de saut de transformation, et dans lequel, à condition que la valeur des informations de saut de transformation activé et la valeur des informations de codage de palette soient toutes deux égales à 0, les informations de paramètre de quantification minimum ne sont pas incluses dans les informations d'image,

20 dans lequel, dans le cas où le saut de transformation serait appliqué au bloc actuel, un paramètre de quantification pour le bloc actuel est déterminé sur la base des informations de paramètre de quantification minimum, et dans lequel, dans le cas où le mode palette serait appliqué au bloc actuel, le paramètre de quantification pour le bloc actuel est déterminé sur la base des informations de paramètre de quantification minimum.

- 25 4. Procédé de transmission de données d'image, pour le codage vidéo versatile, VVC, le procédé de transmission comprenant:

30 la génération d'un flux binaire en exécutant le procédé de codage d'image selon la revendication 2; et la transmission des données comprenant le flux binaire.

30

35

40

45

50

55

FIG. 1

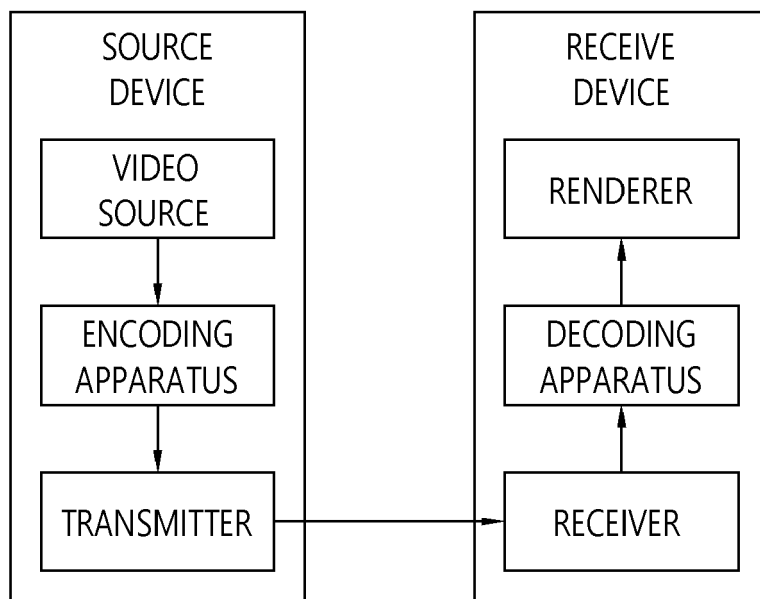


FIG. 2

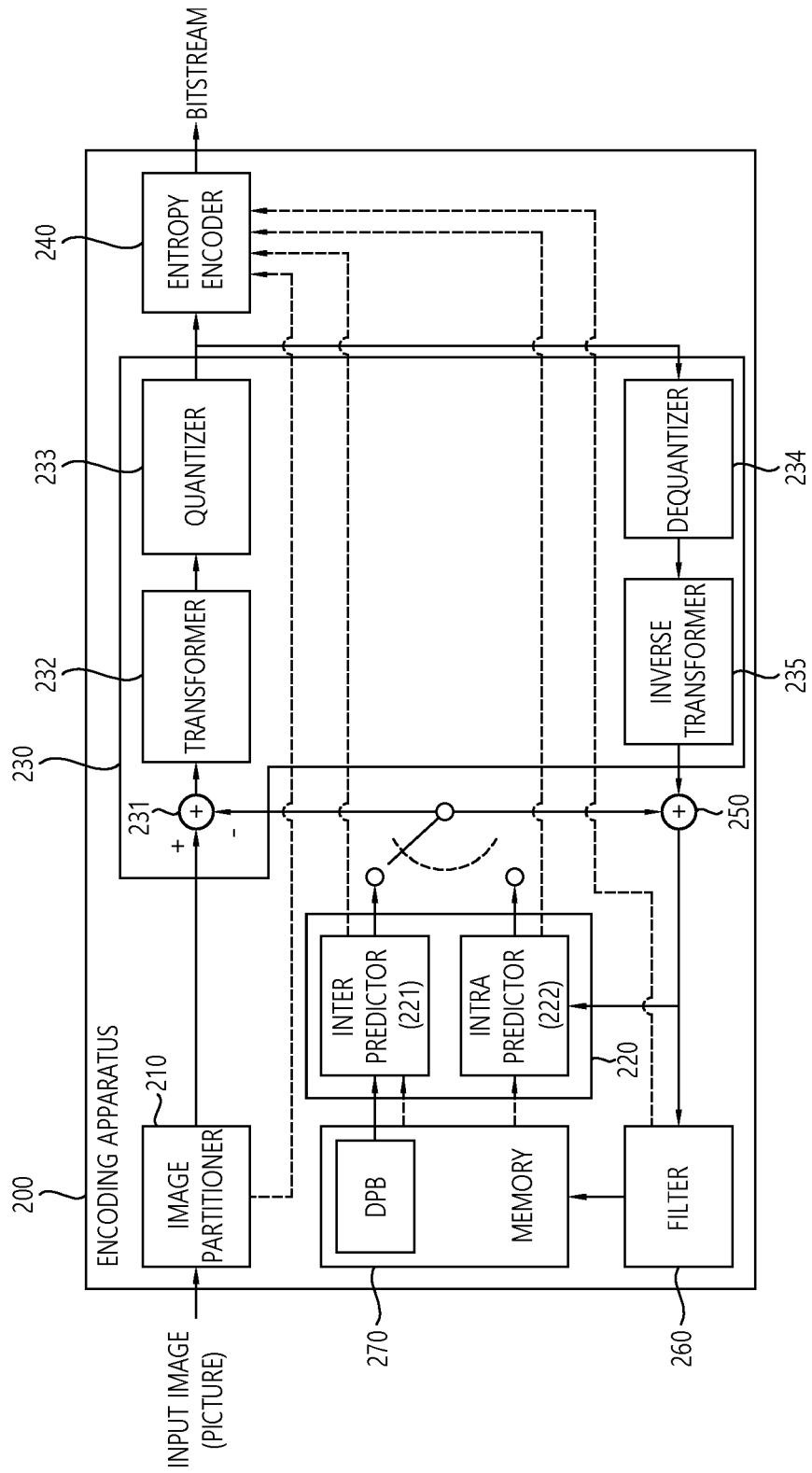


FIG. 3

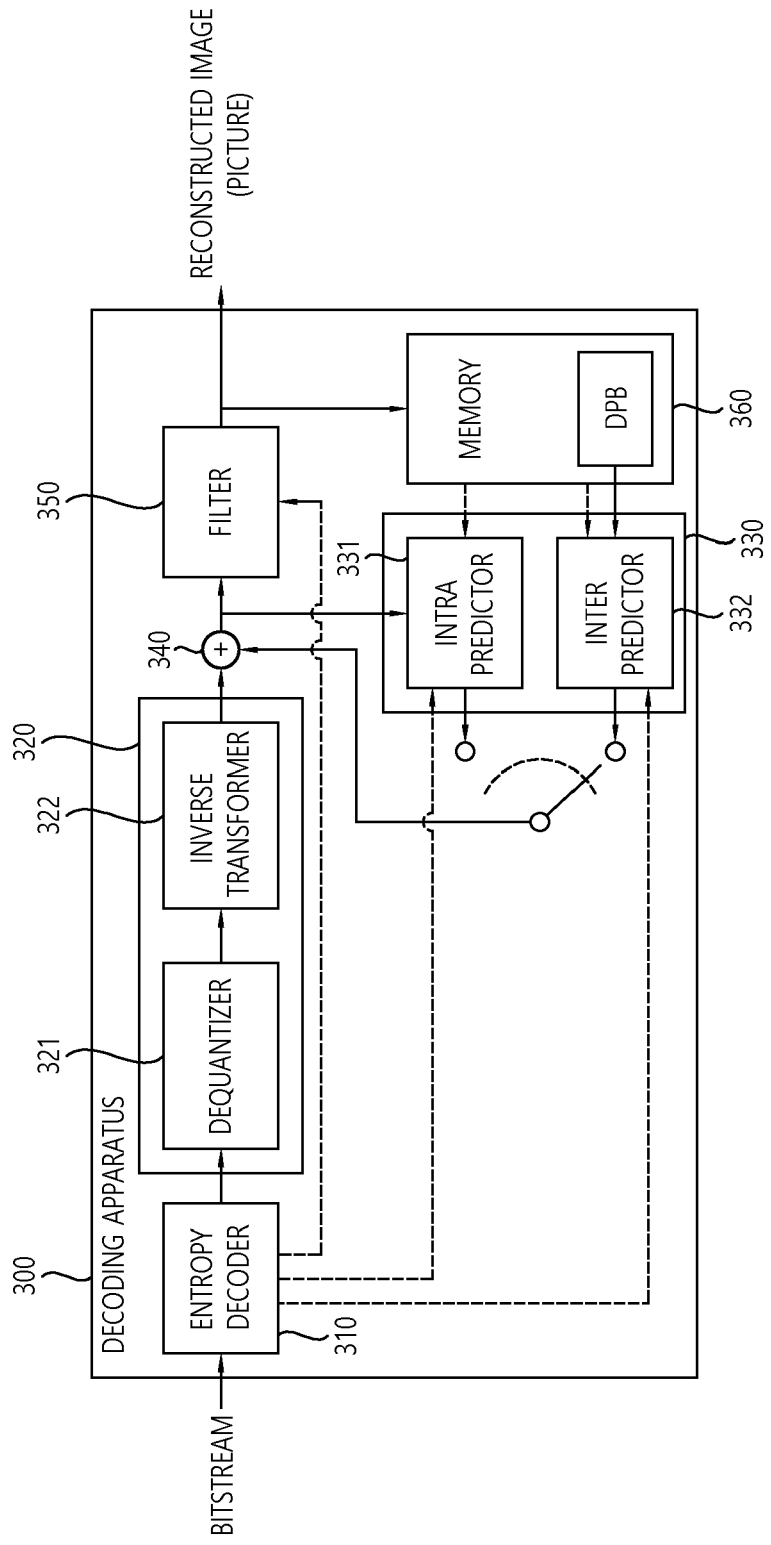


FIG. 4

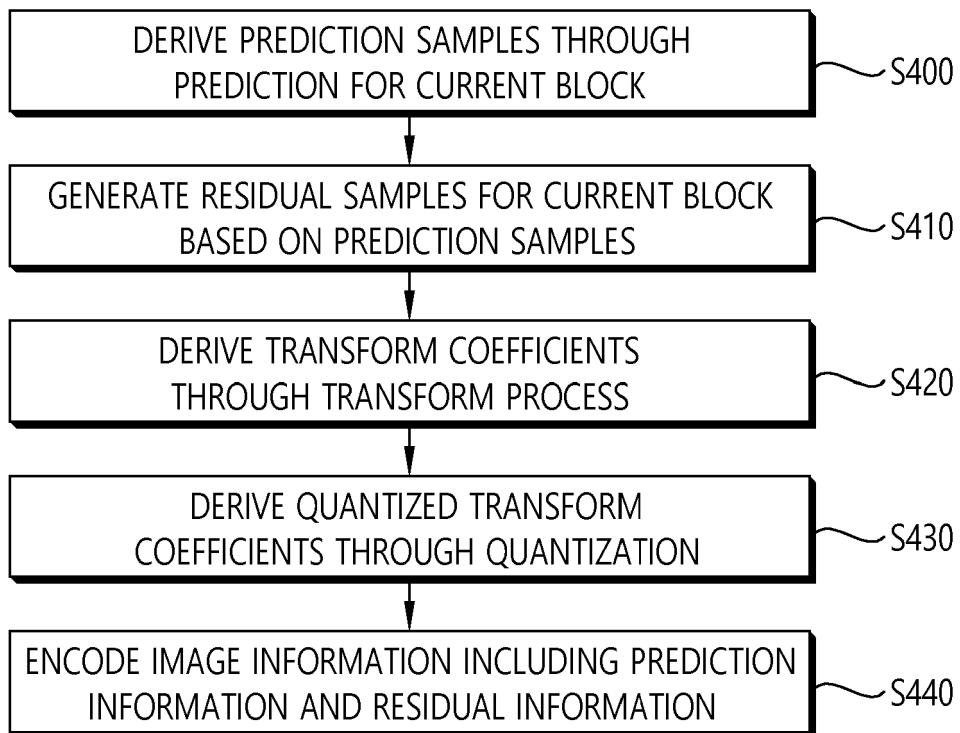


FIG. 5

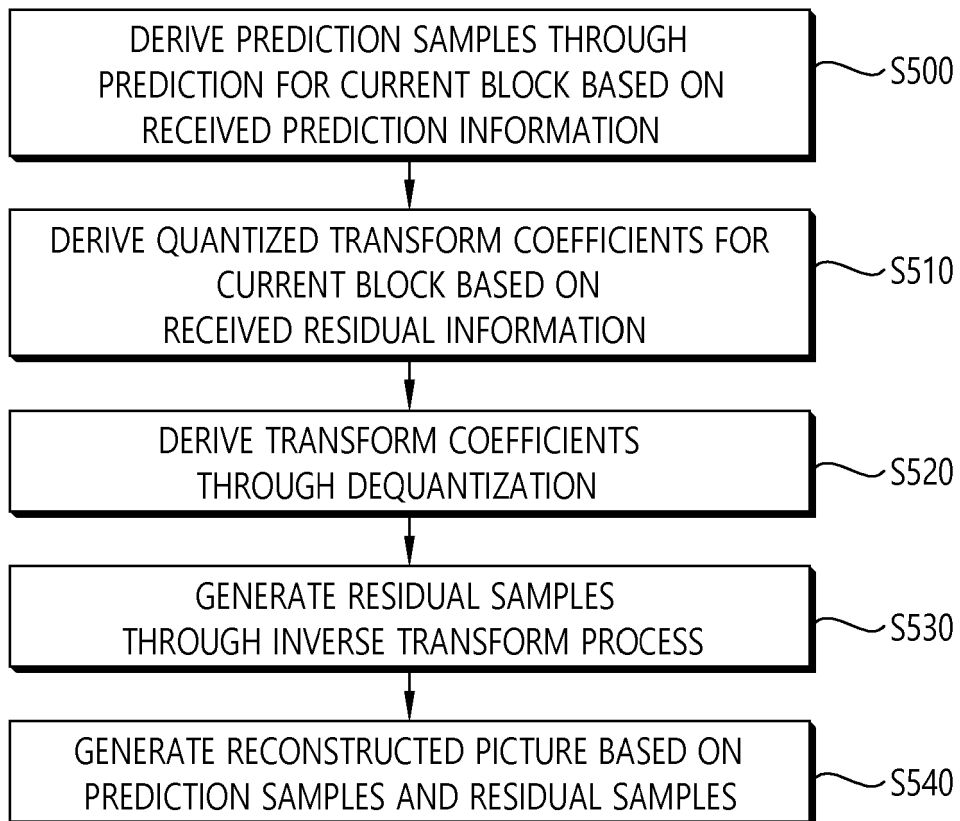


FIG. 6

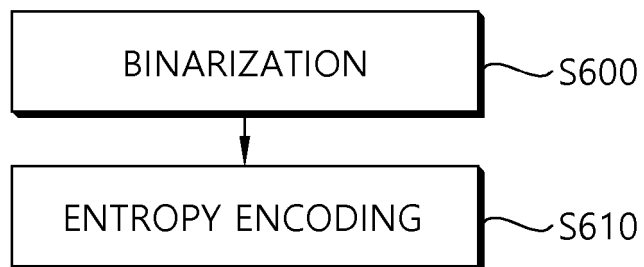


FIG. 7

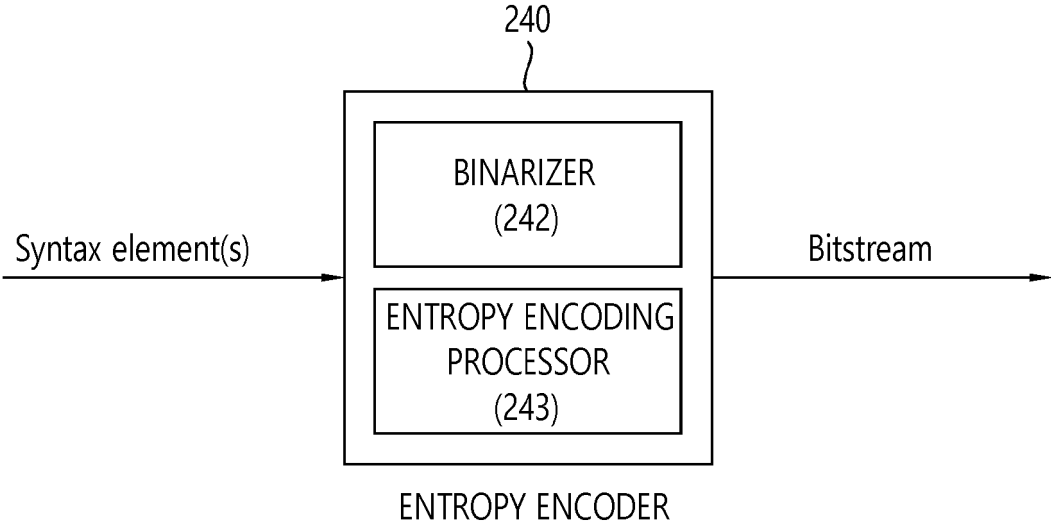


FIG. 8

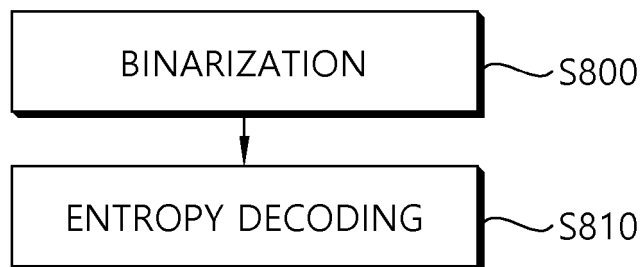


FIG. 9

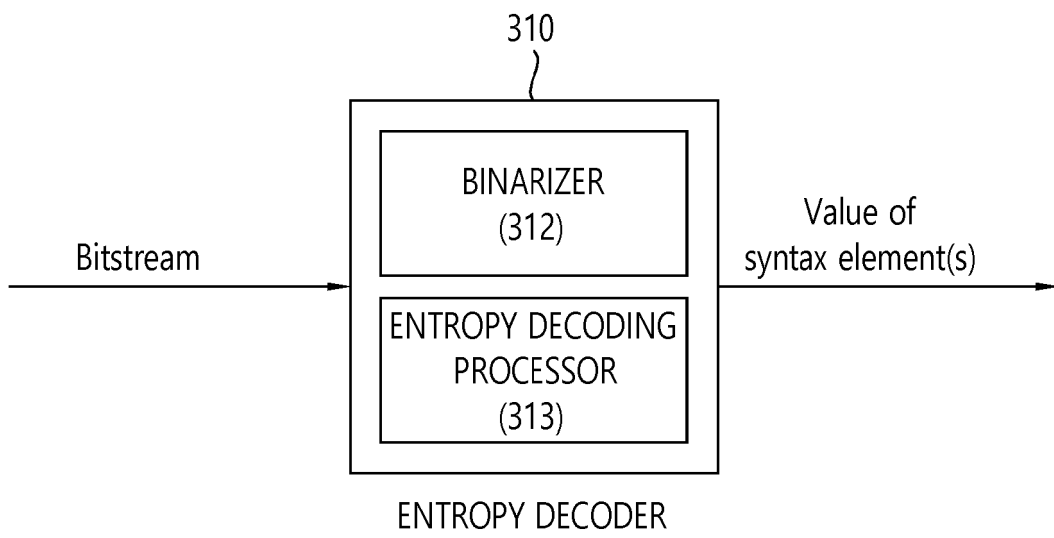


FIG. 10

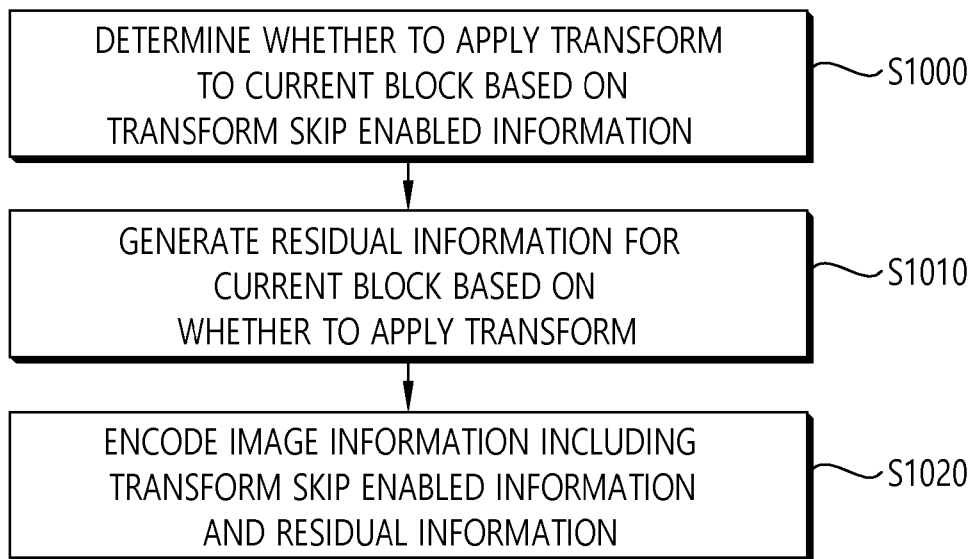


FIG. 11

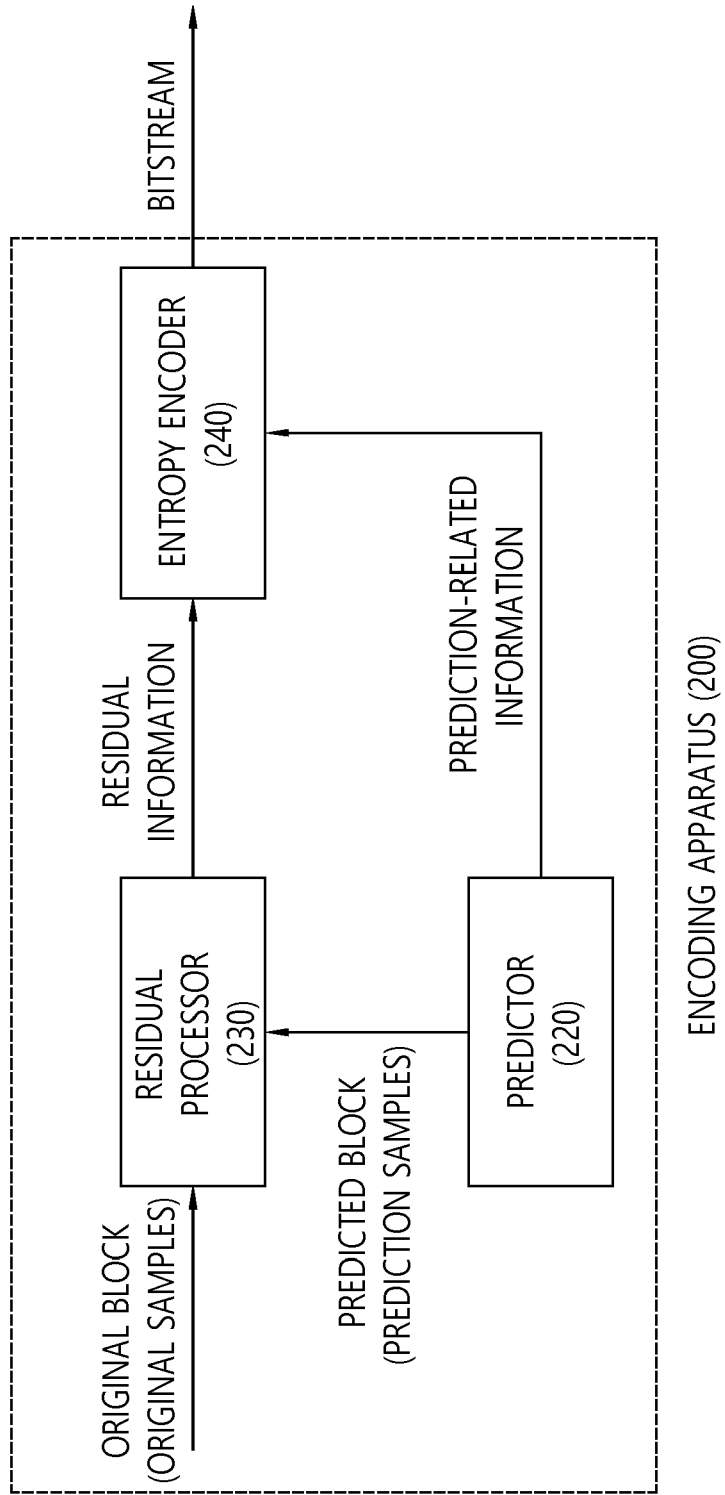


FIG. 12

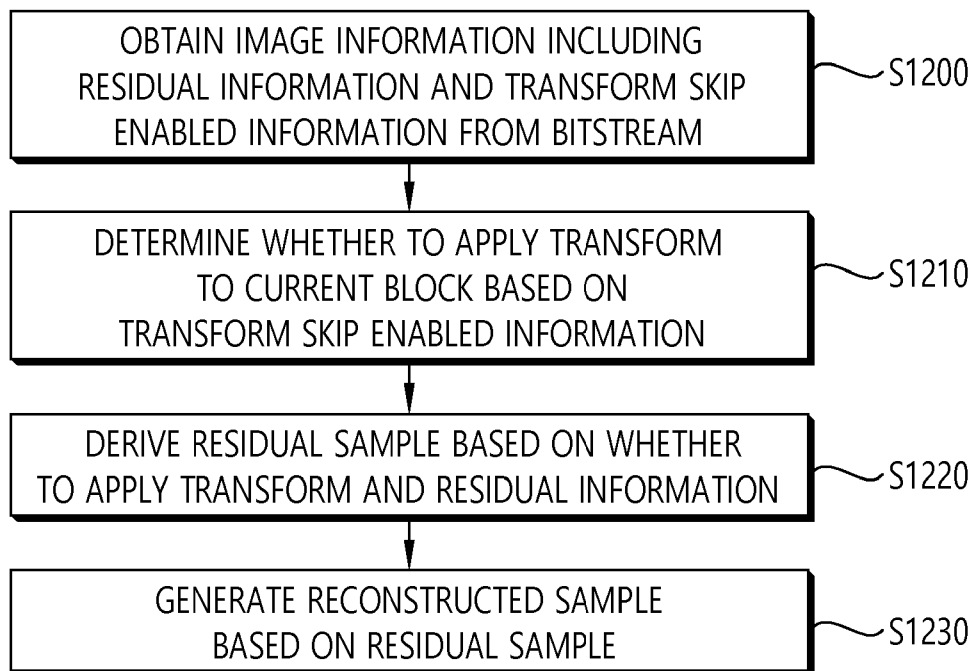


FIG. 13

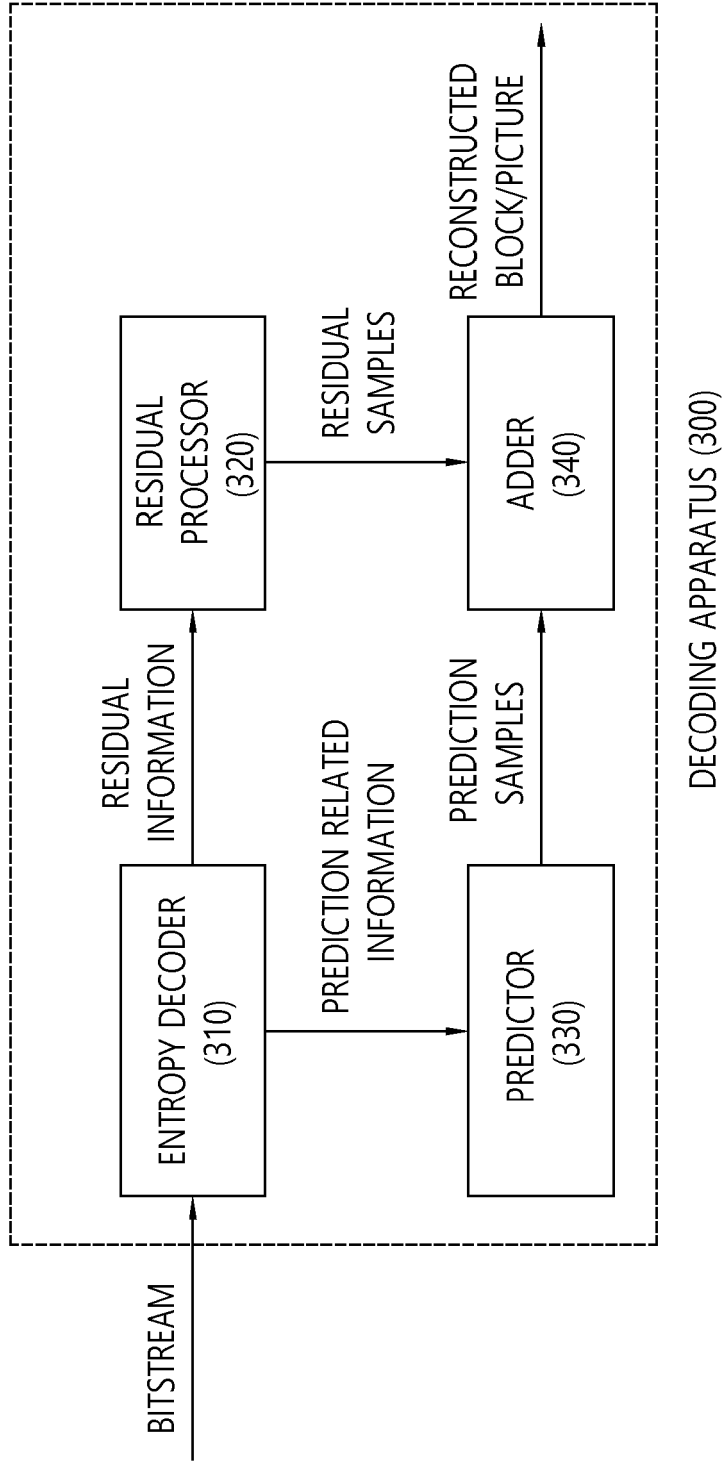
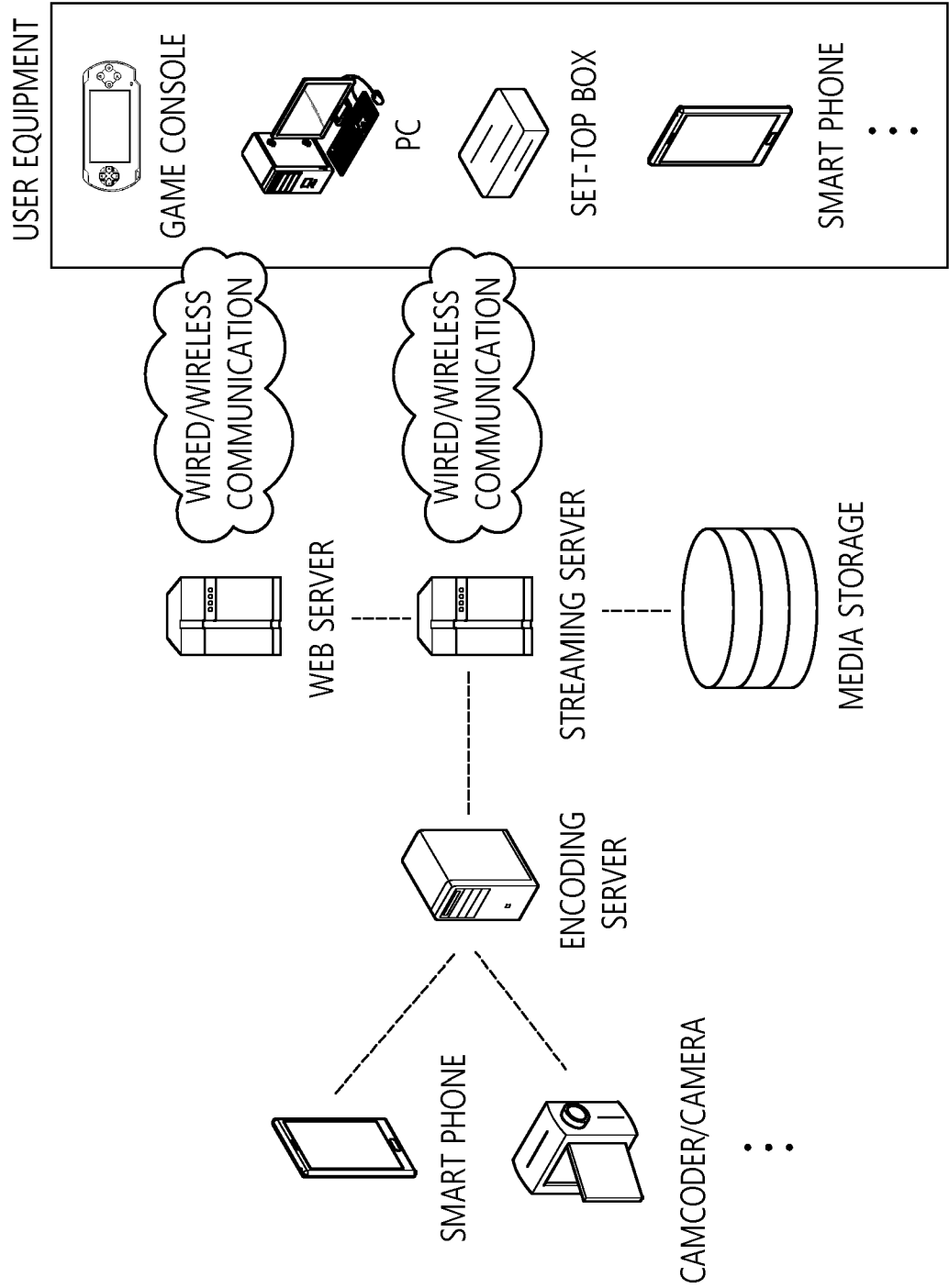


FIG. 14



**REFERENCES CITED IN THE DESCRIPTION**

*This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.*

**Non-patent literature cited in the description**

- **BROSS B et al.** Versatile Video Coding (Draft 6). *JVET MEETING; 20190703 - 20190712; GOTHENBURG; (THE JOINT VIDEO EXPLORATION TEAM OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16)*, 31 July 2019, vol. 15 (JVET-02001-vE), 1-455, [https://jvet-experts.org/doc\\_end\\_user/documents/15\\_Gothenburg/wg11/JVET-02001-v14.zip](https://jvet-experts.org/doc_end_user/documents/15_Gothenburg/wg11/JVET-02001-v14.zip) JVET-02001-vE.docx **[0005]**