

【Invention Title】

PARAMETER UPDATE METHOD FOR ENTROPY CODING AND DECODING OF
CONVERSION COEFFICIENT LEVEL, AND ENTROPY CODING DEVICE AND
ENTROPY DECODING DEVICE OF CONVERSION COEFFICIENT LEVEL USING
5 SAME

【Technical Field】

The present invention relates to video encoding and decoding, and more
particularly, to methods and APPARATUSES for updating a parameter used in
10 entropy encoding and decoding of size information of a transformation coefficient.

【Background Art】

According to image compression methods such as MPEG-1, MPEG-2, or
MPEG-4 H.264/MPEG-4 advanced video coding (AVC), an image is split into blocks
15 having a predetermined size, and then, residual data of the blocks is obtained by
inter prediction or intra prediction. Residual data is compressed by transformation,
quantization, scanning, run length coding, and entropy coding. In entropy coding, a
syntax element such as a transformation coefficient or a motion vector is entropy
encoded to output a bit stream. At a decoder's end, a syntax element is extracted
20 from a bit stream, and decoding is performed based on the extracted syntax
element.

【Disclosure】

【Technical Problem】

25 The present invention provides a method of updating a parameter whereby an
abrupt variation of a parameter used in entropy encoding and decoding of a
transformation coefficient level is gradually varied while preventing an abrupt
variation of said parameter.

The present invention also provides a method of updating a parameter used
30 in binarizing a syntax element such as a transformation coefficient level by using a
binarization method such as a Golomb-rice method or a concatenate code method.

【Technical Solution】

According to the embodiments of the present invention, a parameter updating

method in which a parameter used in binarization of a transformation coefficient level is gradually updated is provided.

【Advantageous Effects】

According to the embodiments of the present invention, by gradually varying a parameter used in entropy encoding of level information of a transformation coefficient, a bit amount generated during encoding may be reduced, and a gain of an image may be increased.

【Description of Drawings】

FIG. 1 is a block diagram of an apparatus for encoding a video, according to an embodiment of the present invention;

FIG. 2 is a block diagram of an apparatus for decoding a video, according to an embodiment of the present invention;

FIG. 3 is a diagram for describing a concept of coding units according to an embodiment of the present invention;

FIG. 4 is a block diagram of a video encoder based on coding units having a hierarchical structure, according to an embodiment of the present invention;

FIG. 5 is a block diagram of a video decoder based on coding units having a hierarchical structure, according to an embodiment of the present invention;

FIG. 6 is a diagram illustrating deeper coding units according to depths, and partitions, according to an embodiment of the present invention;

FIG. 7 is a diagram for describing a relationship between a coding unit and transformation units, according to an embodiment of the present invention;

FIG. 8 is a diagram for describing encoding information of coding units corresponding to a coded depth, according to an embodiment of the present invention;

FIG. 9 is a diagram of deeper coding units according to depths, according to an embodiment of the present invention;

FIGS. 10 through 12 are diagrams for describing a relationship between coding units, prediction units, and frequency transformation units, according to an embodiment of the present invention;

FIG. 13 is a diagram for describing a relationship between a coding unit, a prediction unit, and a transformation unit, according to encoding mode information of Table 1;

FIG. 14 is a flowchart illustrating an operation of entropy encoding and decoding of transformation coefficient information included in a transformation unit, according to an embodiment of the present invention;

FIG. 15 illustrates a transformation unit that is entropy encoded according to an embodiment of the present invention;

FIG. 16 illustrates a significance map corresponding to the transformation unit of FIG. 15 according to an embodiment of the present invention;

FIG. 17 illustrates coeff_abs_level_greater1_flag corresponding to the 4x4 transformation unit of FIG. 15;

FIG. 18 illustrates coeff_abs_level_greater2_flag corresponding to the 4x4 transformation unit of FIG. 15;

FIG. 19 illustrates coeff_abs_level_remaining corresponding to the 4x4 transformation unit of FIG. 15;

FIG. 20 illustrates a table showing syntax elements related to the transformation units illustrated in FIGS. 15 through 19;

FIG. 21 illustrates another example of coeff_abs_level_remaining that is binarized according to an embodiment of the present invention;

FIG. 22 is a block diagram illustrating a structure of an entropy encoding apparatus according to an embodiment of the present invention;

FIG. 23 is a block diagram illustrating a structure of a binarization apparatus according to an embodiment of the present invention;

FIG. 24 is a flowchart illustrating a method of entropy encoding of syntax elements indicating a transformation coefficient level according to an embodiment of the present invention;

FIG. 25 is a block diagram illustrating an entropy decoding apparatus according to an embodiment of the present invention;

FIG. 26 is a block diagram illustrating a structure of a de-binarization apparatus according to an embodiment of the present invention; and
FIG. 27 is a flowchart illustrating a method of entropy decoding of a transformation coefficient level according to an embodiment of the present invention.

【Best Mode】

According to an aspect of the present invention, there is provided a method of updating a parameter for entropy decoding of a transformation coefficient level, the method comprising: parsing transformation coefficient level syntax elements
5 indicating sizes of transformation coefficients included in a transformation unit, from a bit stream; determining whether to renew a previous parameter by comparing a size of a previous transformation coefficient that is restored before a current transformation coefficient, with a predetermined critical value obtained based on the previous parameter that is used in de-binarization of a previous transformation
10 coefficient level syntax element indicating the size of the previous transformation coefficient; obtaining a parameter used in de-binarization of a current transformation coefficient level syntax element indicating a size of the current transformation coefficient by renewing or maintaining the previous parameter based on a result of the determining; and obtaining the size of the current transformation coefficient by
15 de-binarizing the current transformation coefficient level syntax element by using the obtained parameter, wherein the predetermined critical value is set to have a value proportional to the previous parameter, and when the previous parameter is renewed, the renewed parameter has a value that gradually increases compared to the previous parameter.

20 According to another aspect of the present invention, there is provided an apparatus for entropy decoding a transformation coefficient level, the apparatus comprising: a parsing unit that parses transformation coefficient level syntax elements indicating sizes of transformation coefficients included in a transformation unit, from a bit stream; a parameter determining unit that determines whether to renew a
25 previous parameter by comparing a size of a previous transformation coefficient that is restored before a current transformation coefficient, with a predetermined critical value obtained based on the previous parameter that is used in de-binarization of a previous transformation coefficient level syntax element indicating a size of the previous transformation coefficient, and that obtains a parameter used in
30 de-binarization of a current transformation coefficient level syntax element indicating a size of the current transformation coefficient by renewing or maintaining the previous parameter based on a result of the determination; a syntax element restoring unit that obtains the size of the current transformation coefficient by de-binarizing the current transformation coefficient level syntax element by using the

obtained parameter, wherein the predetermined critical value is set to have a value proportional to the previous parameter, and when the previous parameter is renewed, the renewed parameter has a value that gradually increases compared to the previous parameter. According to another aspect of the present invention, there is provided a method of updating a parameter for entropy encoding of a transformation coefficient level, the method comprising: obtaining, in a predetermined scanning order, transformation coefficient level syntax elements indicating sizes of transformation coefficients included in a transformation unit; determining whether to renew a previous parameter by comparing a size of a previous transformation coefficient that is encoded before a current transformation coefficient, with a predetermined critical value obtained based on the previous parameter that is used in binarization of a previous transformation coefficient level syntax element indicating the size of the previous transformation coefficient; obtaining a parameter used in binarization of a current transformation coefficient level syntax element indicating a size of the current transformation coefficient by renewing or maintaining the previous parameter based on a result of the determination; and outputting a bit string corresponding to a transformation coefficient level syntax element of the current transformation coefficient by binarizing the transformation coefficient level syntax element of the current transformation coefficient by using the obtained parameter, wherein the predetermined critical value is set to have a value proportional to the previous parameter, and when the previous parameter is renewed, the renewed parameter has a value that gradually increases compared to the previous parameter.

According to another aspect of the present invention, there is provided an apparatus for entropy encoding a transformation coefficient level, the apparatus comprising: a parameter determining unit that obtains, in a predetermined scanning order, transformation coefficient level syntax elements indicating sizes of transformation coefficients included in a transformation unit, determines whether to renew a previous parameter by comparing a size of a previous transformation coefficient that is encoded before a current transformation coefficient, with a predetermined critical value obtained based on the previous parameter that is used in binarization of a previous transformation coefficient level syntax element indicating the size of the previous transformation coefficient, and obtains a parameter used in binarization of a current transformation coefficient level syntax element indicating a size of the current transformation coefficient by renewing or maintaining the previous

parameter based on a result of the determination; a bit string generating unit that outputs bit strings corresponding to the transformation coefficient level syntax element of the current transformation coefficient by binarizing the transformation coefficient level syntax elements of the current transformation coefficient by using the obtained parameter, wherein the predetermined critical value is set to have a value proportional to the previous parameter, and when the previous parameter is renewed, the renewed parameter has a value that gradually increases compared to the previous parameter.

10 **【Mode for Invention】**

Hereinafter, a method and apparatus for updating a parameter used in entropy encoding and decoding size information of a transformation unit according to an embodiment of the present invention will be described with reference to FIGS. 1 through 13. In addition, a method of entropy encoding and decoding a syntax element obtained by using the method of entropy encoding and decoding of a video described with reference to FIGS. 1 through 13 will be described in detail with reference to FIGS. 14 through 29. Expressions such as “at least one of,” when preceding a list of elements, modify the entire list of elements and do not modify the individual elements of the list.

20 FIG. 1 is a block diagram of a video encoding apparatus 100 according to an embodiment of the present invention.

The video encoding apparatus 100 includes a hierarchical encoder 110 and an entropy encoder 120.

The hierarchical encoder 110 may split a current picture to be encoded, in units of predetermined data units to perform encoding on each of the data units. In detail, the hierarchical encoder 110 may split a current picture based on a maximum coding unit, which is a coding unit of a maximum size. The maximum coding unit according to an embodiment of the present invention may be a data unit having a size of 32x32, 64x64, 128x128, 256x256, etc., wherein a shape of the data unit is a square which has width and length in squares of 2 and is greater than 8.

A coding unit according to an embodiment of the present invention may be characterized by a maximum size and a depth. The depth denotes the number of times the coding unit is spatially split from the maximum coding unit, and as the depth deepens, deeper encoding units according to depths may be split from the

maximum coding unit to a minimum coding unit. A depth of the maximum coding unit is an uppermost depth and a depth of the minimum coding unit is a lowermost depth. Since a size of a coding unit corresponding to each depth decreases as the depth of the maximum coding unit deepens, a coding unit corresponding to an upper
5 depth may include a plurality of coding units corresponding to lower depths.

As described above, image data of the current picture is split into the maximum coding units according to a maximum size of the coding unit, and each of the maximum coding units may include deeper coding units that are split according to depths. Since the maximum coding unit according to an embodiment of the
10 present invention is split according to depths, the image data of a spatial domain included in the maximum coding unit may be hierarchically classified according to depths.

A maximum depth and a maximum size of a coding unit, which limit the total number of times a height and a width of the maximum coding unit are hierarchically
15 split, may be predetermined.

The hierarchical encoder 110 encodes at least one split region obtained by splitting a region of the maximum coding unit according to depths, and determines a depth to output finally encoded image data according to the at least one split region. In other words, the hierarchical encoder 110 determines a coded depth by encoding
20 the image data in the deeper coding units according to depths, according to the maximum coding unit of the current picture, and selecting a depth having the least encoding error. The determined coded depth and the encoded image data according to maximum encoding units are output to the entropy encoder 120.

The image data in the maximum coding unit is encoded based on the deeper
25 coding units corresponding to at least one depth equal to or smaller than the maximum depth, and results of encoding the image data are compared based on each of the deeper coding units. A depth having the least encoding error may be selected after comparing encoding errors of the deeper coding units. At least one coded depth may be selected for each maximum coding unit.

The size of the maximum coding unit is split as a coding unit is hierarchically
30 split according to depths and as the number of coding units increases. Also, even if coding units correspond to a same depth in one maximum coding unit, it is determined whether to split each of the coding units corresponding to the same depth to a lower depth by measuring an encoding error of the image data of each

coding unit, separately. Accordingly, even when image data is included in one maximum coding unit, the image data is split into regions according to the depths, and the encoding errors may differ according to regions in the one maximum coding unit, and thus the coded depths may differ according to regions in the image data.

Thus, one or more coded depths may be determined in one maximum coding unit, and the image data of the maximum coding unit may be divided according to coding units of at least one coded depth.

Accordingly, the hierarchical encoder 110 may determine coding units having a tree structure included in the maximum coding unit. The 'coding units having a tree structure' according to an embodiment of the present invention include coding units corresponding to a depth determined to be the coded depth, from among all deeper coding units included in the maximum coding unit. A coding unit having a coded depth may be hierarchically determined according to depths in the same region of the maximum coding unit, and may be independently determined in different regions. Similarly, a coded depth in a current region may be independently determined from a coded depth in another region.

A maximum depth according to an embodiment of the present invention is an index related to the number of times splitting is performed from a maximum coding unit to a minimum coding unit. A first maximum depth according to an embodiment of the present invention may denote the total number of times splitting is performed from the maximum coding unit to the minimum coding unit. A second maximum depth according to an embodiment of the present invention may denote the total number of depth levels from the maximum coding unit to the minimum coding unit. For example, when a depth of the maximum coding unit is 0, a depth of a coding unit, in which the maximum coding unit is split once, may be set to 1, and a depth of a coding unit, in which the maximum coding unit is split twice, may be set to 2. Here, if the minimum coding unit is a coding unit in which the maximum coding unit is split four times, five depth levels of depths 0, 1, 2, 3, and 4 exist, and thus the first maximum depth may be set to 4, and the second maximum depth may be set to 5.

Prediction encoding and transformation may be performed according to the maximum coding unit. The prediction encoding and the transformation are also performed based on the deeper coding units according to a depth equal to or depths less than the maximum depth, according to the maximum coding unit.

Since the number of deeper coding units increases whenever the maximum

coding unit is split according to depths, encoding including the prediction encoding and the transformation is performed on all of the deeper coding units generated as the depth deepens. For convenience of description, the prediction encoding and the transformation will now be described based on a coding unit of a current depth,
5 in a maximum coding unit.

The video encoding apparatus 100 may variously select a size or shape of a data unit for encoding the image data. In order to encode the image data, operations, such as prediction encoding, transformation, and entropy encoding, are performed, and at this time, the same data unit may be used for all operations or
10 different data units may be used for each operation.

For example, the video encoding apparatus 100 may select not only a coding unit for encoding the image data, but also a data unit different from the coding unit so as to perform the prediction encoding on the image data in the coding unit.

In order to perform prediction encoding in the maximum coding unit, the prediction encoding may be performed based on a coding unit corresponding to a
15 coded depth, i.e., based on a coding unit that is no longer split into coding units corresponding to a lower depth. Hereinafter, the coding unit that is no longer split and becomes a basis unit for prediction encoding will now be referred to as a 'prediction unit'. A partition obtained by splitting the prediction unit may include a
20 prediction unit or a data unit obtained by splitting at least one of a height and a width of the prediction unit.

For example, when a coding unit of $2N \times 2N$ (where N is a positive integer) is no longer split and becomes a prediction unit of $2N \times 2N$, a size of a partition may be $2N \times 2N$, $2N \times N$, $N \times 2N$, or $N \times N$. Examples of a partition type include symmetrical
25 partitions that are obtained by symmetrically splitting a height or width of the prediction unit, partitions obtained by asymmetrically splitting the height or width of the prediction unit, such as $1:n$ or $n:1$, partitions that are obtained by geometrically splitting the prediction unit, and partitions having arbitrary shapes.

A prediction mode of the prediction unit may be at least one of an intra mode, an inter mode, and a skip mode. For example, the intra mode or the inter mode
30 may be performed on the partition of $2N \times 2N$, $2N \times N$, $N \times 2N$, or $N \times N$. Also, the skip mode may be performed only on the partition of $2N \times 2N$. The encoding is independently performed on one prediction unit in a coding unit, thereby selecting a prediction mode having the least encoding error.

The video encoding apparatus 100 may also perform the transformation on the image data in a coding unit based not only on the coding unit for encoding the image data, but also based on a data unit that is different from the coding unit.

In order to perform the transformation in the coding unit, the transformation may be performed based on a data unit having a size smaller than or equal to the coding unit. For example, the data unit for the transformation may include a data unit for an intra mode and a data unit for an inter mode.

A data unit used as a base of the transformation will now be referred to as a 'transformation unit'. Similarly to the coding unit, the transformation unit in the coding unit may be recursively split into smaller sized regions, so that the transformation unit may be determined independently in units of regions. Thus, residual data in the coding unit may be divided according to the transformation unit having the tree structure according to transformation depths.

A transformation depth indicating the number of times splitting is performed to reach the transformation unit by splitting the height and width of the coding unit may also be set in the transformation unit. For example, in a current coding unit of $2N \times 2N$, a transformation depth may be 0 when the size of a transformation unit is $2N \times 2N$, may be 1 when the size of a transformation unit is $N \times N$, and may be 2 when the size of a transformation unit is $N/2 \times N/2$. That is, the transformation unit having the tree structure may also be set according to transformation depths.

Encoding information according to coding units corresponding to a coded depth requires not only information about the coded depth, but also about information related to prediction encoding and transformation. Accordingly, the hierarchical encoder 110 not only determines a coded depth having the least encoding error, but also determines a partition type in a prediction unit, a prediction mode according to prediction units, and a size of a transformation unit for transformation.

Coding units according to a tree structure in a maximum coding unit and a method of determining a partition, according to embodiments of the present invention, will be described in detail later with reference to FIGS. 3 through 12.

The hierarchical encoder 110 may measure an encoding error of deeper coding units according to depths by using Rate-Distortion Optimization based on Lagrangian multipliers.

The entropy encoder 120 outputs the image data of the maximum coding unit,

which is encoded based on the at least one coded depth determined by the hierarchical encoder 110, and information about the encoding mode according to the coded depth, in bit streams. The encoded image data may be a coding result of residual data of an image. The information about the encoding mode according to the coded depth may include information about the coded depth, information about the partition type in the prediction unit, prediction mode information, and size information of the transformation unit. In particular, as will be described later, when entropy encoding a syntax element indicating a size of a transformation unit, the entropy encoder 120 binarizes a syntax element indicating a size of a transformation but according to bit strings by using the parameter which is gradually updated. The operation of entropy encoding a transformation unit by using the entropy encoding unit 120 will be described later.

The information about the coded depth may be defined by using split information according to depths, which indicates whether encoding is performed on coding units of a lower depth instead of a current depth. If the current depth of the current coding unit is the coded depth, image data in the current coding unit is encoded and output, and thus the split information may be defined not to split the current coding unit to a lower depth. Alternatively, if the current depth of the current coding unit is not the coded depth, the encoding is performed on the coding unit of the lower depth, and thus the split information may be defined to split the current coding unit to obtain the coding units of the lower depth.

If the current depth is not the coded depth, encoding is performed on the coding unit that is split into the coding unit of the lower depth. Since at least one coding unit of the lower depth exists in one coding unit of the current depth, the encoding is repeatedly performed on each coding unit of the lower depth, and thus the encoding may be recursively performed for the coding units having the same depth.

Since the coding units having a tree structure are determined for one maximum coding unit, and information about at least one encoding mode is determined for a coding unit of a coded depth, information about at least one encoding mode may be determined for one maximum coding unit. Also, a coded depth of the image data of the maximum coding unit may be different according to locations since the image data is hierarchically split according to depths, and thus information about the coded depth and the encoding mode may be set for the image

data.

Accordingly, the entropy encoder 120 may assign encoding information about a corresponding coded depth and an encoding mode to at least one of the coding unit, the prediction unit, and a minimum unit included in the maximum coding unit.

5 The minimum unit according to an embodiment of the present invention is a square-shaped data unit obtained by splitting the minimum coding unit constituting the lowermost depth by 4. Alternatively, the minimum unit may be a maximum square-shaped data unit that may be included in all of the coding units, prediction units, partition units, and transformation units included in the maximum coding unit.

10 For example, the encoding information output through the entropy encoder 120 may be classified into encoding information according to coding units and encoding information according to prediction units. The encoding information according to the coding units may include the information about the prediction mode and about the size of the partitions. The encoding information according to the
15 prediction units may include information about an estimated direction of an inter mode, about a reference image index of the inter mode, about a motion vector, about a chroma component of an intra mode, and about an interpolation method of the intra mode. Also, information about a maximum size of the coding unit defined according to pictures, slices, or GOPs, and information about a maximum depth may
20 be inserted into a header of a bit stream.

In the video encoding apparatus 100, the deeper coding unit may be a coding unit obtained by dividing a height or width of a coding unit of an upper depth, which is one layer above, by two. In other words, when the size of the coding unit of the current depth is $2N \times 2N$, the size of the coding unit of the lower depth is $N \times N$. Also,
25 the coding unit of the current depth having the size of $2N \times 2N$ may include a maximum number of four coding units of the lower depth.

Accordingly, the video encoding apparatus 100 may form the coding units having the tree structure by determining coding units having an optimum shape and an optimum size for each maximum coding unit, based on the size of the maximum
30 coding unit and the maximum depth determined considering characteristics of the current picture. Also, since encoding may be performed on each maximum coding unit by using any one of various prediction modes and transformations, an optimum encoding mode may be determined considering characteristics of the coding unit of various image sizes.

Thus, if an image having a high resolution or a large data amount is encoded in a conventional macroblock, a number of macroblocks per picture excessively increases. Accordingly, a number of pieces of compressed information generated for each macroblock increases, and thus it is difficult to transmit the compressed information and data compression efficiency decreases. However, by using the video encoding apparatus 100, image compression efficiency may be increased since a coding unit is adjusted while considering characteristics of an image while increasing a maximum size of a coding unit while considering a size of the image.

FIG. 2 is a block diagram of a video decoding apparatus 200 according to an embodiment of the present invention.

The video decoding apparatus 200 includes a parser 210, an entropy decoder 220, and a hierarchical decoder 230. Definitions of various terms, such as a coding unit, a depth, a prediction unit, a transformation unit, and information about various encoding modes, for various operations of the video decoding apparatus 200 are identical to those described with reference to FIG. 1 and the video encoding apparatus 100.

The parser 210 receives a bit stream of an encoded video to parse a syntax element. The entropy decoder 220 extracts syntax elements indicating encoded image data based on coding units having a structure by performing entropy decoding of parsed syntax elements, and outputs the extracted syntax elements to the hierarchical decoder 230. That is, the entropy decoder 220 performs entropy decoding of syntax elements that are received in the form of bit strings of 0 and 1, thereby restoring the syntax elements.

Also, the entropy decoder 220 extracts information about a coded depth, an encoding mode, color component information, prediction mode information, etc. for the coding units having a tree structure according to each maximum coding unit, from the parsed bitstream. The extracted information about the coded depth and the encoding mode is output to the hierarchical decoder 230. The image data in a bit stream is split into the maximum coding unit so that the hierarchical decoder 230 may decode the image data for each maximum coding unit.

The information about the coded depth and the encoding mode according to the maximum coding unit may be set for information about at least one coding unit corresponding to the coded depth, and information about an encoding mode may include information about a partition type of a corresponding coding unit

corresponding to the coded depth, about a prediction mode, and a size of a transformation unit. Also, splitting information according to depths may be extracted as the information about the coded depth.

The information about the coded depth and the encoding mode according to each maximum coding unit extracted by the entropy decoder 220 is information about a coded depth and an encoding mode determined to generate a minimum encoding error when an encoder, such as the video encoding apparatus 100, repeatedly performs encoding for each deeper coding unit according to depths according to each maximum coding unit. Accordingly, the video decoding apparatus 200 may restore an image by decoding the image data according to a coded depth and an encoding mode that generates the minimum encoding error.

Since encoding information about the coded depth and the encoding mode may be assigned to a predetermined data unit from among a corresponding coding unit, a prediction unit, and a minimum unit, the entropy decoder 220 may extract the information about the coded depth and the encoding mode according to the predetermined data units. When information about a coded depth and encoding mode of a corresponding maximum coding unit is assigned to each of predetermined data units, the predetermined data units to which the same information about the coded depth and the encoding mode is assigned may be inferred to be the data units included in the same maximum coding unit.

Also, as will be described later, the entropy decoder 220 de-binarizes a syntax element indicating a size of a transformation coefficient by using a parameter that is gradually updated. An operation of obtaining size information of a transformation coefficient by using the entropy decoder 220 by de-binarizing a bit string corresponding to a syntax element indicating a size of a transformation unit will be described in detail later.

The hierarchical decoder 230 restores the current picture by decoding the image data in each maximum coding unit based on the information about the coded depth and the encoding mode according to the maximum coding units. In other words, the hierarchical decoder 230 may decode the encoded image data based on the extracted information about the partition type, the prediction mode, and the transformation unit for each coding unit from among the coding units having the tree structure included in each maximum coding unit. A decoding process may include prediction including intra prediction and motion compensation, and inverse

transformation.

The hierarchical decoder 230 may perform intra prediction or motion compensation according to a partition and a prediction mode of each coding unit, based on the information about the partition type and the prediction mode of the prediction unit of the coding unit according to coded depths.

Also, the hierarchical decoder 230 may perform inverse transformation according to each transformation unit in the coding unit, based on the information about the size of the transformation unit of the coding unit according to coded depths, so as to perform the inverse transformation according to maximum coding units.

The hierarchical decoder 230 may determine at least one coded depth of a current maximum coding unit by using split information according to depths. If the split information indicates that image data is no longer split in the current depth, the current depth is a coded depth. Accordingly, the hierarchical decoder 230 may decode the coding unit of the current depth with respect to the image data of the current maximum coding unit by using the information about the partition type of the prediction unit, the prediction mode, and the size of the transformation unit.

In other words, data units containing the encoding information including the same split information may be gathered by observing the encoding information set assigned for the predetermined data unit from among the coding unit, the prediction unit, and the minimum unit, and the gathered data units may be considered to be one data unit to be decoded by the hierarchical decoder 230 in the same encoding mode.

The video decoding apparatus 200 may obtain information about at least one coding unit that generates the minimum encoding error when encoding is recursively performed for each maximum coding unit, and may use the information to decode the current picture. In other words, encoded image data of the coding units having the tree structure determined to be the optimum coding units in each maximum coding unit may be decoded.

Accordingly, even if image data has a high resolution and a large amount of data, the image data may be efficiently decoded and restored by using a size of a coding unit and an encoding mode, which are adaptively determined according to characteristics of the image data, by using information about an optimum encoding mode received from an encoder.

A method of determining coding units having a tree structure, a prediction unit,

and a transformation unit, according to an embodiment of the present invention, will now be described with reference to FIGS. 3 through 13.

FIG. 3 is a diagram for describing a concept of coding units according to an embodiment of the present invention.

5 A size of a coding unit may be expressed in width x height, and may be 64x64, 32x32, 16x16, and 8x8. A coding unit of 64x64 may be split into partitions of 64x64, 64x32, 32x64, or 32x32; and a coding unit of 32x32 may be split into partitions of 32x32, 32x16, 16x32, or 16x16; a coding unit of 16x16 may be split into partitions of 16x16, 16x8, 8x16, or 8x8; and a coding unit of 8x8 may be split into partitions of 8x8,
10 8x4, 4x8, or 4x4.

Regarding video data 310, a resolution of 1920x1080, a maximum size of a coding unit of 64, and a maximum depth of 2 are set. Regarding video data 320, a resolution of 1920x1080, a maximum size of a coding unit of 64, and a maximum depth of 3 are set. Regarding video data 330, a resolution of 352x288, a maximum
15 size of a coding unit of 16, and a maximum depth of 1 are set. The maximum depth shown in FIG. 3 denotes a total number of splits from a maximum coding unit to a minimum coding unit.

If a resolution is high or a data amount is large, a maximum size of a coding unit may be large so as to not only increase encoding efficiency but also to
20 accurately reflect characteristics of an image. Accordingly, the maximum size of the coding unit of the video data 310 and 320 having the higher resolution than the video data 330 may be 64.

Since the maximum depth of the video data 310 is 2, coding units 315 of the video data 310 may include a maximum coding unit having a long axis size of 64, and
25 coding units having long axis sizes of 32 and 16 since depths are deepened to two layers by splitting the maximum coding unit twice. Meanwhile, since the maximum depth of the video data 330 is 1, coding units 335 of the video data 330 may include a maximum coding unit having a long axis size of 16, and coding units having a long axis size of 8 since depths are deepened to one layer by splitting the maximum
30 coding unit once.

Since the maximum depth of the video data 320 is 3, coding units 325 of the video data 320 may include a maximum coding unit having a long axis size of 64, and coding units having long axis sizes of 32, 16, and 8 since the depths are deepened to 3 layers by splitting the maximum coding unit three times. As a depth

deepens, detailed information may be precisely expressed.

FIG. 4 is a block diagram of a video encoder 400 based on coding units having a hierarchical structure, according to an embodiment of the present invention.

An intra predictor 410 performs intra prediction on coding units in an intra mode, with respect to a current frame 405, and a motion estimator 420 and a motion compensator 425 respectively perform inter estimation and motion compensation on coding units in an inter mode by using the current frame 405 and a reference frame 495.

Data output from the intra predictor 410, the motion estimator 420, and the motion compensator 425 is output as a quantized transformation coefficient through a transformer 430 and a quantizer 440. The quantized transformation coefficient is restored as data in a spatial domain through an inverse quantizer 460 and an inverse transformer 470, and the restored data in the spatial domain is output as the reference frame 495 after being post-processed through a deblocking unit 480 and a loop filtering unit 490. The quantized transformation coefficient may be output as a bitstream 455 through an entropy encoder 450.

The entropy encoding unit 450 arithmetically encodes syntax elements related to a transformation unit such as a significance map indicating a position of a transformation unit that is not 0, a first critical value flag (coeff_abs_level_greater1_flag) indicating whether a transformation unit has a value greater than 1 or not, a second critical value flag (coeff_abs_level_greater2_flag) indicating whether a transformation unit has a value greater than 2, and a size information of a transformation coefficient (coeff_abs_level_remaining) corresponding to a difference between a basic level (baseLevel) that is determined based on the first critical value flag and the second critical value flag and a real transformation coefficient (abscoeff).

In order for the video encoder 400 to be applied in the video encoding apparatus 100, all elements of the video encoder 400, i.e., the intra predictor 410, the motion estimator 420, the motion compensator 425, the transformer 430, the quantizer 440, the entropy encoder 450, the inverse quantizer 460, the inverse transformer 470, the deblocking unit 480, and the loop filtering unit 490, have to perform operations based on each coding unit from among coding units having a tree structure while considering the maximum depth of each maximum coding unit.

Specifically, the intra predictor 410, the motion estimator 420, and the motion

compensator 425 determine partitions and a prediction mode of each coding unit from among the coding units having a tree structure while considering the maximum size and the maximum depth of a current maximum coding unit, and the transformer 430 determines the size of the transformation unit in each coding unit from among the coding units having a tree structure.

FIG. 5 is a block diagram of a video decoder 500 based on coding units, according to an embodiment of the present invention.

A parser 510 parses encoded image data to be decoded and information about encoding required for decoding, from a bitstream 505. The encoded image data passes through the decoder 520 and the inverse quantizer 530 to be output as inversely quantized data. The entropy decoder 520 obtains elements related to a transformation unit from a bit stream, that is, a significance map indicating a position of a transformation unit that is not 0, a first critical value flag (coeff_abs_level_greater1_flag) indicating whether a transformation unit has a value greater than 1 or not, a second critical value flag (coeff_abs_level_greater2_flag) indicating whether a transformation unit has a value greater than 2, and a size information of a transformation coefficient (coeff_abs_level_remaining) corresponding to a difference between a basic level (baseLevel) that is determined based on the first critical value flag and the second critical value flag and a real transformation coefficient (abscoeff), and arithmetically decodes the obtained syntax elements so as to restore the syntax elements.

An inverse transformer 540 restores the inversely quantized data to image data in a spatial domain. An intra predictor 550 performs intra prediction on coding units in an intra mode with respect to the image data in the spatial domain, and a motion compensator 560 performs motion compensation on coding units in an inter mode by using a reference frame 585.

The image data in the spatial domain, which has passed through the intra predictor 550 and the motion compensator 560, may be output as a restored frame 595 after being post-processed through a deblocking unit 570 and a loop filtering unit 580. Also, the image data, which is post-processed through the deblocking unit 570 and the loop filtering unit 580, may be output as the reference frame 585.

In order for the video decoder 500 to be applied in the video decoding apparatus 200, all elements of the video decoder 500, i.e., the parser 510, the entropy decoder 520, the inverse quantizer 530, the inverse transformer 540, the

intra predictor 550, the motion compensator 560, the deblocking unit 570, and the loop filtering unit 580, perform operations based on coding units having a tree structure for each maximum coding unit.

The intra predictor 550 and the motion compensator 560 determine a partition and a prediction mode for each coding unit having a tree structure, and the inverse transformer 540 has to determine a size of a transformation unit for each coding unit.

FIG. 6 is a diagram illustrating deeper coding units according to depths, and partitions, according to an embodiment of the present invention.

The video encoding apparatus 100 and the video decoding apparatus 200 use hierarchical coding units so as to consider characteristics of an image. A maximum height, a maximum width, and a maximum depth of coding units may be adaptively determined according to the characteristics of the image, or may be differently set by a user. Sizes of deeper coding units according to depths may be determined according to the predetermined maximum size of the coding unit.

In a hierarchical structure 600 of coding units according to an embodiment of the present invention, the maximum height and the maximum width of the coding units are each 64, and the maximum depth is 4. Since a depth deepens along a vertical axis of the hierarchical structure 600, a height and a width of the deeper coding unit are each split. Also, a prediction unit and partitions, which are bases for prediction encoding of each deeper coding unit, are shown along a horizontal axis of the hierarchical structure 600.

In other words, a coding unit 610 is a maximum coding unit in the hierarchical structure 600, wherein a depth is 0 and a size, i.e., a height by width, is 64x64. The depth deepens along the vertical axis, and a coding unit 620 having a size of 32x32 and a depth of 1, a coding unit 630 having a size of 16x16 and a depth of 2, a coding unit 640 having a size of 8x8 and a depth of 3, and a coding unit 650 having a size of 4x4 and a depth of 4 exist. The coding unit 650 having the size of 4x4 and the depth of 4 is a minimum coding unit.

The prediction unit and the partitions of a coding unit are arranged along the horizontal axis according to each depth. In other words, if the coding unit 610 having the size of 64x64 and the depth of 0 is a prediction unit, the prediction unit may be split into partitions included in the encoding unit 610, i.e. a partition 610 having a size of 64x64, partitions 612 having the size of 64x32, partitions 614 having

the size of 32x64, or partitions 616 having the size of 32x32.

Similarly, a prediction unit of the coding unit 620 having the size of 32x32 and the depth of 1 may be split into partitions included in the coding unit 620, i.e. a partition 620 having a size of 32x32, partitions 622 having a size of 32x16, partitions 624 having a size of 16x32, and partitions 626 having a size of 16x16.

Similarly, a prediction unit of the coding unit 630 having the size of 16x16 and the depth of 2 may be split into partitions included in the coding unit 630, i.e. a partition having a size of 16x16 included in the coding unit 630, partitions 632 having a size of 16x8, partitions 634 having a size of 8x16, and partitions 636 having a size of 8x8.

Similarly, a prediction unit of the coding unit 640 having the size of 8x8 and the depth of 3 may be split into partitions included in the coding unit 640, i.e. a partition having a size of 8x8 included in the coding unit 640, partitions 642 having a size of 8x4, partitions 644 having a size of 4x8, and partitions 646 having a size of 4x4.

The coding unit 650 having the size of 4x4 and the depth of 4 is the minimum coding unit and a coding unit of the lowermost depth. A prediction unit of the coding unit 650 is only assigned to a partition having a size of 4x4.

In order to determine the at least one coded depth of the coding units constituting the maximum coding unit 610, the hierarchical encoder 110 of the video encoding apparatus 100 performs encoding for coding units corresponding to each depth included in the maximum coding unit 610.

The number of deeper coding units according to depths including data in the same range and the same size increases as the depth deepens. For example, four coding units corresponding to a depth of 2 are required to cover data that is included in one coding unit corresponding to a depth of 1. Accordingly, in order to compare encoding results of the same data according to depths, the coding unit corresponding to the depth of 1 and four coding units corresponding to the depth of 2 are each encoded.

In order to perform encoding for a current depth from among the depths, a least encoding error may be selected for the current depth by performing encoding for each prediction unit in the coding units corresponding to the current depth, along the horizontal axis of the hierarchical structure 600. Alternatively, the minimum encoding error may be searched for by comparing the least encoding errors

according to depths and performing encoding for each depth as the depth deepens along the vertical axis of the hierarchical structure 600. A depth and a partition having the minimum encoding error in the maximum coding unit 610 may be selected as the coded depth and a partition type of the maximum coding unit 610.

5 FIG. 7 is a diagram for describing a relationship between a coding unit 710 and transformation units 720, according to an embodiment of the present invention.

The video encoding apparatus 100 or the video decoding apparatus 200 encodes or decodes an image according to coding units having sizes smaller than or equal to a maximum coding unit for each maximum coding unit. Sizes of
10 transformation units for transformation during encoding may be selected based on data units that are not larger than a corresponding coding unit.

For example, in the video encoding apparatus 100 or the video decoding apparatus 200, if a size of the coding unit 710 is 64x64, transformation may be performed by using the transformation units 720 having a size of 32x32.

15 Also, data of the coding unit 710 having the size of 64x64 may be encoded by performing the transformation on each of the transformation units having the size of 32x32, 16x16, 8x8, and 4x4, which are smaller than 64x64, and then a transformation unit having the least coding error may be selected.

FIG. 8 is a diagram for describing encoding information of coding units
20 corresponding to a coded depth, according to an embodiment of the present invention.

An output unit 130 of the video encoding apparatus 100 may encode and transmit information 800 about a partition type, information 810 about a prediction mode, and information 820 about a size of a transformation unit for each coding unit
25 corresponding to a coded depth, as information about an encoding mode.

The information 800 indicates information about a shape of a partition obtained by splitting a prediction unit of a current coding unit, wherein the partition is a data unit for prediction encoding the current coding unit. For example, a current coding unit CU_0 having a size of 2Nx2N may be split into any one of a partition 802
30 having a size of 2Nx2N, a partition 804 having a size of 2NxN, a partition 806 having a size of Nx2N, and a partition 808 having a size of NxN. Here, the information 800 about a partition type is set to indicate one of the partition 802 having a size of 2Nx2N, the partition 804 having a size of 2NxN, the partition 806 having a size of Nx2N, and the partition 808 having a size of NxN.

The information 810 indicates a prediction mode of each partition. For example, the information 810 may indicate a mode of prediction encoding performed on a partition indicated by the information 800, i.e., an intra mode 812, an inter mode 814, or a skip mode 816.

5 The information 820 indicates a transformation unit to be based on when transformation is performed on a current coding unit. For example, the transformation unit may be a first intra transformation unit 822, a second intra transformation unit 824, a first inter transformation unit 826, or a second inter transformation unit 828.

10 The image data and encoding data extracting unit 210 of the video decoding apparatus 200 may extract and use the information 800 information about coding units, the information 810 about a prediction mode, and the information 820 about a size of a transformation unit, for decoding, according to each deeper coding unit.

15 FIG. 9 is a diagram of deeper coding units according to depths, according to an embodiment of the present invention.

Split information may be used to indicate a change of a depth. The split information indicates whether a coding unit of a current depth is split into coding units of a lower depth.

20 A prediction unit 910 for prediction encoding of a coding unit 900 having a depth of 0 and a size of $2N_0 \times 2N_0$ may include partitions of a partition type 912 having a size of $2N_0 \times 2N_0$, a partition type 914 having a size of $2N_0 \times N_0$, a partition type 916 having a size of $N_0 \times 2N_0$, and a partition type 918 having a size of $N_0 \times N_0$. FIG. 9 only illustrates the partition types 912 through 918 which are obtained by symmetrically splitting the prediction unit 910, but a partition type is not
25 limited thereto, and the partitions of the prediction unit 910 may include asymmetrical partitions, partitions having a predetermined shape, and partitions having a geometrical shape.

30 Prediction encoding is repeatedly performed on one partition having a size of $2N_0 \times 2N_0$, two partitions having a size of $2N_0 \times N_0$, two partitions having a size of $N_0 \times 2N_0$, and four partitions having a size of $N_0 \times N_0$, according to each partition type. The prediction encoding in an intra mode and an inter mode may be performed on the partitions having the sizes of $2N_0 \times 2N_0$, $N_0 \times 2N_0$, $2N_0 \times N_0$, and $N_0 \times N_0$. The prediction encoding in a skip mode is performed only on the partition having the size of $2N_0 \times 2N_0$.

If an encoding error is the smallest in one of the partition types 912 through 916 having the sizes of $2N_0 \times 2N_0$, $2N_0 \times N_0$, and $N_0 \times 2N_0$, the prediction unit 910 may not be split into a lower depth.

If the encoding error is the smallest in the partition type 918 having the size of $N_0 \times N_0$, a depth is changed from 0 to 1 to split the partition type 918 in operation 920, and encoding is repeatedly performed on partition type coding units having a depth of 2 and a size of $N_0 \times N_0$ to search for a minimum encoding error.

A prediction unit 940 for prediction encoding of the (partition type) coding unit 930 having a depth of 1 and a size of $2N_1 \times 2N_1 (=N_0 \times N_0)$ may include partitions of a partition type 942 having a size of $2N_1 \times 2N_1$, a partition type 944 having a size of $2N_1 \times N_1$, a partition type 946 having a size of $N_1 \times 2N_1$, and a partition type 948 having a size of $N_1 \times N_1$.

If an encoding error is the smallest in the partition type 948 having the size of $N_1 \times N_1$, a depth is changed from 1 to 2 to split the partition type 948 in operation 950, and encoding is repeatedly performed on coding units 960, which have a depth of 2 and a size of $N_2 \times N_2$ to search for a minimum encoding error.

When a maximum depth is d , a split operation according to each depth may be performed up to when a depth becomes $d-1$, and split information may be encoded as up to when a depth is one of 0 to $d-2$. In other words, when encoding is performed up to when the depth is $d-1$ after a coding unit corresponding to a depth of $d-2$ is split in operation 970, a prediction unit 990 for prediction encoding a coding unit 980 having a depth of $d-1$ and a size of $2N_{(d-1)} \times 2N_{(d-1)}$ may include partitions of a partition type 992 having a size of $2N_{(d-1)} \times 2N_{(d-1)}$, a partition type 994 having a size of $2N_{(d-1)} \times N_{(d-1)}$, a partition type 996 having a size of $N_{(d-1)} \times 2N_{(d-1)}$, and a partition type 998 having a size of $N_{(d-1)} \times N_{(d-1)}$.

Prediction encoding may be repeatedly performed on one partition having a size of $2N_{(d-1)} \times 2N_{(d-1)}$, two partitions having a size of $2N_{(d-1)} \times N_{(d-1)}$, two partitions having a size of $N_{(d-1)} \times 2N_{(d-1)}$, four partitions having a size of $N_{(d-1)} \times N_{(d-1)}$ from among the partition types 992 through 998 to search for a partition type having a minimum encoding error.

Even when the partition type 998 having the size of $N_{(d-1)} \times N_{(d-1)}$ has the minimum encoding error, since a maximum depth is d , a coding unit $CU_{(d-1)}$ having a depth of $d-1$ is no longer split to a lower depth, and a coded depth for the coding units constituting the current maximum coding unit 900 is determined to be $d-1$ and

a partition type of the current maximum coding unit 900 may be determined to be $N_{(d-1)} \times N_{(d-1)}$. Also, since the maximum depth is d , split information for the minimum coding unit 952 is not set.

A data unit 999 may be a 'minimum unit' for the current maximum coding unit.

5 A minimum unit according to an embodiment of the present invention may be a rectangular data unit obtained by splitting the minimum coding unit 980 by 4. By performing the encoding repeatedly, the video encoding apparatus 100 may select a depth having the least encoding error by comparing encoding errors according to depths of the coding unit 900 to determine a coded depth, and set a corresponding
10 partition type and a prediction mode as an encoding mode of the coded depth.

As such, the minimum encoding errors according to depths are compared in all of the depths of 1 through d , and a depth having the least encoding error may be determined as a coded depth. The coded depth, the partition type of the prediction unit, and the prediction mode may be encoded and transmitted as information about
15 an encoding mode. Also, since a coding unit is split from a depth of 0 to a coded depth, only split information of the coded depth is set to 0, and split information of depths excluding the coded depth is set to 1.

The entropy decoder 220 of the video decoding apparatus 200 may extract and use the information about the coded depth and the prediction unit of the coding
20 unit 900 to decode the coding unit 912. The video decoding apparatus 200 may determine a depth, in which split information is 0, as a coded depth by using split information according to depths, and use information about an encoding mode of the corresponding depth for decoding.

FIGS. 10 through 12 are diagrams for describing a relationship between
25 coding units 1010, prediction units 1060, and transformation units 1070 according to an embodiment of the present invention.

The coding units 1010 are coding units having a tree structure, corresponding to coded depths determined by the video encoding apparatus 100, in a maximum coding unit. The prediction units 1060 are partitions of prediction units of each of
30 the coding units 1010, and the transformation units 1070 are transformation units of each of the coding units 1010.

When a depth of a maximum coding unit is 0 in the coding units 1010, depths of coding units 1012 and 1054 are 1, depths of coding units 1014, 1016, 1018, 1028, 1050, and 1052 are 2, depths of coding units 1020, 1022, 1024, 1026, 1030, 1032,

and 1048 are 3, and depths of coding units 1040, 1042, 1044, and 1046 are 4.

In the prediction units 1060, some coding units 1014, 1016, 1022, 1032, 1048, 1050, 1052, and 1054 are obtained by splitting the coding units. In other words, partition types in the coding units 1014, 1022, 1050, and 1054 have a size of $2N \times N$, partition types in the coding units 1016, 1048, and 1052 have a size of $N \times 2N$, and a partition type of the coding unit 1032 has a size of $N \times N$. Prediction units and partitions of the coding units 1010 are smaller than or equal to each coding unit.

Transformation or inverse transformation is performed on image data of the coding unit 1052 in the transformation units 1070 in a data unit that is smaller than the coding unit 1052. Also, the coding units 1014, 1016, 1022, 1032, 1048, 1050, 1052, and 1054 in the transformation units 1070 are different from those in the prediction units 1060 in terms of sizes and shapes. In other words, the video encoding apparatus 100 and the video decoding apparatus 200 may perform intra prediction, motion estimation, motion compensation, transformation, and inverse transformation individually on a data unit in the same coding unit.

Accordingly, encoding is recursively performed on each of coding units having a hierarchical structure in each region of a maximum coding unit to determine an optimum coding unit, and thus coding units having a recursive tree structure may be obtained. Encoding information may include split information about a coding unit, information about a partition type, information about a prediction mode, and information about a size of a transformation unit.

Table 1 shows the encoding information that may be set by the video encoding apparatus 100 and the video decoding apparatus 200.

Table 1

Split Information 0 (Encoding on Coding Unit having Size of $2N \times 2N$ and Current Depth of d)					Split Information 1
Prediction Mode	Partition Type		Size of Transformation Unit		Repeatedly Encode Coding Units Having Lower Depth of $d+1$
Intra Inter Skip (Only	Symmetrical Partition Type	Asymmetrical Partition Type	Split Information 0 of Transformation Unit	Split Information 1 of Transformation Unit	

2Nx2N)	2NxN 2NxN nLx2N nRx2N	2NxN 2NxN nLx2N nRx2N	2Nx2N	NxN (Symmetrical Partition Type) N/2xN/2 (Asymmetrical Partition Type)	
--------	--------------------------------	--------------------------------	-------	---	--

The entropy encoder 120 of the video encoding apparatus 100 may output the encoding information about the coding units having a tree structure, and the entropy decoder 220 of the video decoding apparatus 200 may extract the encoding information about the coding units having a tree structure from a received bitstream.

Split information indicates whether a current coding unit is split into coding units of a lower depth. If split information of a current depth d is 0, a depth, in which a current coding unit is no longer split into a lower depth, is a coded depth, and thus information about a partition type, a prediction mode, and a size of a transformation unit may be defined for the coded depth. If the current coding unit is further split according to the split information, encoding is independently performed on four split coding units of a lower depth.

A prediction mode may be one of an intra mode, an inter mode, and a skip mode. The intra mode and the inter mode may be defined in all partition types, and the skip mode is defined only in a partition type having a size of 2Nx2N.

The information about the partition type may indicate symmetrical partition types having sizes of 2Nx2N, 2NxN, Nx2N, and NxN, which are obtained by symmetrically splitting a height or a width of a prediction unit, and asymmetrical partition types having sizes of 2NxN, 2NxN, nLx2N, and nRx2N, which are obtained by asymmetrically splitting the height or width of the prediction unit. The asymmetrical partition types having the sizes of 2NxN and 2NxN may be respectively obtained by splitting the height of the prediction unit in 1:n and n:1 (where n is an integer greater than 1), and the asymmetrical partition types having the sizes of nLx2N and nRx2N may be respectively obtained by splitting the width of the prediction unit in 1:n and n:1.

The size of the transformation unit may be set to be two types in the intra mode and two types in the inter mode. In other words, if split information of the transformation unit is 0, the size of the transformation unit may be 2Nx2N, which is the size of the current coding unit. If split information of the transformation unit is 1,

the transformation units may be obtained by splitting the current coding unit. Also, if a partition type of the current coding unit having the size of $2N \times 2N$ is a symmetrical partition type, a size of a transformation unit may be $N \times N$, and if the partition type of the current coding unit is an asymmetrical partition type, the size of the transformation unit may be $N/2 \times N/2$.

The encoding information about coding units having a tree structure may include at least one of a coding unit corresponding to a coded depth, a prediction unit, and a minimum unit. The coding unit corresponding to the coded depth may include at least one of a prediction unit and a minimum unit containing the same encoding information.

Accordingly, it is determined whether adjacent data units are included in the same coding unit corresponding to the coded depth by comparing encoding information of the adjacent data units. Also, a corresponding coding unit corresponding to a coded depth is determined by using encoding information of a data unit, and thus a distribution of coded depths in a maximum coding unit may be determined.

Accordingly, if a current coding unit is predicted based on encoding information of adjacent data units, encoding information of data units in deeper coding units adjacent to the current coding unit may be directly referred to and used.

Alternatively, if a current coding unit is predicted based on encoding information of adjacent data units, data units adjacent to the current coding unit are searched using encoded information of the data units, and the searched adjacent coding units may be referred to for predicting the current coding unit.

FIG. 13 is a diagram for describing a relationship between a coding unit, a prediction unit, and a transformation unit according to the encoding mode information of Table 1.

A maximum coding unit 1300 includes coding units 1302, 1304, 1306, 1312, 1314, 1316, and 1318 of coded depths. Here, since the coding unit 1318 is a coding unit of a coded depth, split information may be set to 0. Information about a partition type of the coding unit 1318 having a size of $2N \times 2N$ may be set to be one of a partition type 1322 having a size of $2N \times 2N$, a partition type 1324 having a size of $2N \times N$, a partition type 1326 having a size of $N \times 2N$, a partition type 1328 having a size of $N \times N$, a partition type 1332 having a size of $2N \times nU$, a partition type 1334 having a size of $2N \times nD$, a partition type 1336 having a size of $nL \times 2N$, and a partition

type 1338 having a size of $nR \times 2N$.

When the partition type is set to be symmetrical, i.e. the partition type 1322, 1324, 1326, or 1328, a transformation unit 1342 having a size of $2N \times 2N$ is set if split information (TU size flag) of a transformation unit is 0, and a transformation unit 1344 having a size of $N \times N$ is set if a TU size flag is 1.

When the partition type is set to be asymmetrical, i.e., the partition type 1332, 1334, 1336, or 1338, a transformation unit 1352 having a size of $2N \times 2N$ is set if a TU size flag is 0, and a transformation unit 1354 having a size of $N/2 \times N/2$ is set if a TU size flag is 1.

The TU size flag is a type of transformation index; a size of a transformation unit corresponding to a transformation index may be modified according to a prediction unit type or a partition type of a coding unit.

When the partition type is set to be symmetrical, i.e. the partition type 1322, 1324, 1326, or 1328, the transformation unit 1342 having a size of $2N \times 2N$ is set if a TU size flag of a transformation unit is 0, and the transformation unit 1344 having a size of $N \times N$ is set if a TU size flag is 1.

When the partition type is set to be asymmetrical, i.e., the partition type 1332 ($2N \times nU$), 1334 ($2N \times nD$), 1336 ($nL \times 2N$), or 1338 ($nR \times 2N$), the transformation unit 1352 having a size of $2N \times 2N$ is set if a TU size flag is 0, and the transformation unit 1354 having a size of $N/2 \times N/2$ is set if a TU size flag is 1.

Referring to FIG. 9, the TU size flag described above is a flag having a value of 0 or 1, but the TU size flag is not limited to 1 bit, and a transformation unit may be hierarchically split while the TU size flag increases from 0. The transformation unit split information (TU size flag) may be used as an example of a transformation index.

In this case, when a TU size flag according to an embodiment is used with a maximum size and a minimum size of a transformation unit, the size of the actually used transformation unit may be expressed. The video encoding apparatus 100 may encode maximum transformation unit size information, minimum transformation unit size information, and maximum transformation unit split information. The encoded maximum transformation unit size information, minimum transformation unit size information, and maximum transformation unit split information may be inserted into a sequence parameter set (SPS). The video decoding apparatus 200 may use the maximum transformation unit size information, the minimum transformation unit size information, and the maximum transformation unit split information for video

decoding.

For example, (a) if a size of a current coding unit is 64x64 and a maximum transformation unit is 32x32, (a-1) a size of a transformation unit is 32x32 if a TU size flag is 0; (a-2) a size of a transformation unit is 16x16 if a TU size flag is 1; and (a-3) a size of a transformation unit is 8x8 if a TU size flag is 2.

Alternatively, (b) if a size of a current coding unit is 32x32 and a minimum transformation unit is 32x32, (b-1) a size of a transformation unit is 32x32 if a TU size flag is 0, and since the size of a transformation unit cannot be smaller than 32x32, no more TU size flags may be set.

Alternatively, (c) if a size of a current encoding unit is 64x64 and a maximum TU size flag is 1, a TU size flag may be 0 or 1 and no other TU size flags may be set.

Accordingly, when defining a maximum TU size flag as 'MaxTransformSizeIndex', a minimum TU size flag as 'MinTransformSize', and a transformation unit in the case when a TU size flag is 0, that is, a basic transformation unit RootTu as 'RootTuSize', a size of a minimum transformation unit 'CurrMinTuSize', which is available in a current coding unit, may be defined by Equation (1) below.

$$\begin{aligned} &\text{CurrMinTuSize} \\ &= \max (\text{MinTransformSize}, \text{RootTuSize}/(2^{\text{MaxTransformSizeIndex}})) \dots (1) \end{aligned}$$

In comparison with the size of the minimum transformation unit 'CurrMinTuSize' that is available in the current coding unit, the basic transformation unit size 'RootTuSize', which is a size of a transformation unit when if a TU size flag is 0, may indicate a maximum transformation unit which may be selected in regard to a system. That is, according to Equation (1),

'RootTuSize/(2^{MaxTransformSizeIndex})' is a size of a transformation unit that is obtained by splitting 'RootTuSize', which is a size of a transformation unit when transformation unit split information is 0, by the number of splitting times corresponding to the maximum transformation unit split information, and 'MinTransformSize' is a size of a minimum transformation unit, and thus a smaller value of these may be 'CurrMinTuSize' which is the size of the minimum transformation unit that is available in the current coding unit.

The size of the basic transformation unit 'RootTuSize' according to an embodiment of the present invention may vary according to a prediction mode.

For example, if a current prediction mode is an inter mode, RootTuSize may

be determined according to Equation (2) below. In Equation (2), 'MaxTransformSize' refers to a maximum transformation unit size, and 'PUSize' refers to a current prediction unit size.

$$\text{RootTuSize} = \min(\text{MaxTransformSize}, \text{PUSize}) \dots\dots\dots (2)$$

In other words, if a current prediction mode is an inter mode, the size of the basic transformation unit size 'RootTuSize', which is a transformation unit if a TU size flag is 0, may be set to a smaller value from among the maximum transformation unit size and the current prediction unit size.

If a prediction mode of a current partition unit is an intra mode, 'RootTuSize' may be determined according to Equation (3) below. 'PartitionSize' refers to a size of the current partition unit.

$$\text{RootTuSize} = \min(\text{MaxTransformSize}, \text{PartitionSize}) \dots\dots\dots(3)$$

In other words, if a current prediction mode is an intra mode, the basic transformation unit size 'RootTuSize' may be set to a smaller value from among the maximum transformation unit size and the current partition unit size.

However, it should be noted that the size of the basic transformation unit size 'RootTuSize', which is the current maximum transformation unit size according to an embodiment of the present invention and varies according to a prediction mode of a partition unit, is an example, and factors for determining the current maximum transformation unit size are not limited thereto.

Hereinafter, an entropy encoding operation of a syntax element, which is performed in the entropy encoder 120 of the video encoding apparatus 100 of FIG. 1, and an entropy decoding operation of a syntax element, which is performed in the entropy decoder 220 of the video decoding apparatus 200 of FIG. 2 will be described in detail.

As described above, the video encoding apparatus 100 and the video decoding apparatus 200 perform encoding and decoding by splitting a maximum coding unit into coding units that are smaller than or equal to a maximum coding unit. A prediction unit and a transformation unit used in prediction and transformation may be determined based on costs independently from other data units. Since an optimum coding unit may be determined by recursively encoding each coding unit having a hierarchical structure included in the maximum coding unit, data units having a tree structure may be configured. In other words, for each maximum coding unit, a coding unit having a tree structure, and a prediction unit and a

transformation unit each having a tree structure may be configured. For decoding, hierarchical information, which is information indicating structure information of data units having a hierarchical structure and non-hierarchical information for decoding, besides the hierarchical information, needs to be transmitted.

5 The information related to a hierarchical structure is information needed for determining a coding unit having a tree structure, a prediction unit having a tree structure, and a transformation unit having a tree structure, as described above with reference to FIGS. 10 through 12, and includes a transformation unit split flag (TU size flag) indicating a size of a maximum coding unit, coded depth, partition
10 information of a prediction unit, a split flag indicating whether a coding unit is split or not, size information of a transformation unit, and a transformation unit split flag (TU size flag) indicating whether a transformation unit is split or not. Examples of coding information other than hierarchical structure information include prediction mode information of intra/inter prediction applied to each prediction unit, motion
15 vector information, prediction direction information, color component information applied to each data unit in the case when a plurality of color components are used, and transformation coefficient information. Hereinafter, hierarchical information and extra-hierarchical information may be referred to as a syntax element which is to be entropy encoded or entropy decoded.

20 In particular, according to the embodiments of the present invention, a method of determining a context model for efficiently entropy encoding and decoding a level of a transformation coefficient, that is, size information of syntax elements is provided. Hereinafter, a method of determining a context model for entropy encoding and decoding of a level of a transformation coefficient will be described in
25 detail.

FIG. 14 is a flowchart illustrating an operation of entropy encoding and decoding of transformation coefficient information included in a transformation unit, according to an embodiment of the present invention.

Referring to FIG. 14, coded_block_flag indicating whether a transformation
30 coefficient which is not 0 (hereinafter, referred to as "significant coefficient") exists or not from among transformation coefficients included in a current transformation unit is first entropy encoded or decoded in operation 1410.

If coded_block_flag is 0, there are only transformation coefficients of 0 in the current transformation unit, and thus only a value 0 is entropy encoded or decoded

as coded_block_flag, and transformation coefficient level information is not entropy encoded or decoded.

In operation 1420, if there is a significant coefficient in the current transformation unit, a significance map SigMap indicating a location of a significant coefficient is entropy encoded or decoded.

A significance map SigMap may be formed of a significant bit and predetermined information indicating a location of a last significance coefficient. A significant bit indicates whether a transformation coefficient according to each scan index is a significant coefficient or 0, and may be expressed by significant_coeff_flag[i]. As will be described later, a significance map is set in units of subsets having a predetermined size which is obtained by splitting a transformation unit. Accordingly, significant_coeff_flag[i] indicates whether a transformation coefficient of an i-th scan index from among transformation coefficients included in a subset included in a transformation unit is 0 or not.

According to the conventional H.264, a flag (End-Of-Block) indicating whether each significant coefficient is the last significant coefficient or not is additionally entropy encoded or decoded. However, according to an embodiment of the present invention, location information of the last significant coefficient itself is entropy encoded or decoded. As described above with reference to FIGS. 1 through 13, the size of a transformation unit according to an embodiment of the present invention is not limited to 4x4 but may also be a larger size such as 8x8, 16x16, or 32x32. It is inefficient to additionally entropy encode or decode a flag (End-Of-Block) indicating whether each significant coefficient is the last significant coefficient since the size of the flag (End-Of-Block) increases. Accordingly, according to an embodiment of the present invention, the location information of the last significant coefficient itself may be entropy coded or decoded. For example, if a location of the last significant coefficient is (x, y), where x and y are integers, last_significant_coeff_x and last_significant_coeff_y which are syntax elements indicating coordinate values of (x, y) may be entropy encoded or decoded.

In operation 1430, transformation coefficient level information indicating a size of a transformation coefficient is entropy encoded or decoded. According to the conventional H.264/AVC, level information of a transformation coefficient is expressed by coeff_abs_level_minus1 which is a syntax element. According to the embodiments of the present invention, level information of a transformation

coefficient may be expressed by `coeff_abs_level_greater1_flag` which is a syntax element regarding whether an absolute value of a transformation coefficient is greater than 1, `coeff_abs_level_greater2_flag` which is a syntax element regarding whether an absolute value of a transformation coefficient is greater than 2, and
5 `coeff_abs_level_remaining` which indicates size information of the remaining transformation coefficient.

The syntax element `coeff_abs_level_remaining` indicating the size information of the remaining transformation coefficient has a difference in a range between a size of a transformation coefficient (`absCoeff`) and a basic level value `baseLevel` that
10 is determined by using `coeff_abs_level_greater1_flag` and `coeff_abs_level_greater2_flag`. The basic level value `baseLevel` is determined according to the equation: $\text{baseLevel} = 1 + \text{coeff_abs_level_greather1_flag} + \text{coeff_abs_level_greather2_flag}$, and `coeff_abs_level_remaining` is determined according to the equation: $\text{coeff_abs_level_remaining} = \text{absCoeff} - \text{baseLevel}$. While
15 `coeff_abs_level_greater1_flag` and `coeff_abs_level_greater2_flag` have a value of 0 or 1, the basic level value `baseLevel` may have a value from 1 to 3. Accordingly, `coeff_abs_level_remaining` may be varied from $(\text{absCoeff} - 1)$ to $(\text{absCoeff} - 3)$. As described above, $(\text{absCoeff} - \text{baseLevel})$, which is a difference between the size of an original transformation coefficient `absCoeff` and the basic level value `baseLevel`, is
20 transmitted as size information of a transformation coefficient in order to reduce a size of transmitted data.

FIG. 22 is a block diagram illustrating a structure of an entropy encoding apparatus 2200 according to an embodiment of the present invention. The entropy encoding apparatus 2200 of FIG. 22 corresponds to the entropy encoder 120 of the
25 video encoding apparatus 100 of FIG. 1.

Referring to FIG. 22, the entropy encoding apparatus 2200 includes a binarizer 2210, a context modeler 2220, and a binary arithmetic coder 2230. Also, the binary arithmetic coder 2230 includes a regular coding engine 2232 and a
bypass coding engine 2234.

30 When syntax elements input to the entropy encoding apparatus 2100 are not binary values, the binarizer 2210 binarizes syntax elements so as to output a Bin string consisting of binary values of 0 or 1. Bin denotes each bit of a stream consisting of 0 or 1, and is encoded by context adaptive binary arithmetic coding (CABAC). If a syntax element is data including 0 and 1 at the same frequencies,

the syntax element is output to the bypass coding engine 2234, which does not use a probability, to be encoded.

In particular, the binarizer 2210 binarizes `coeff_abs_level_remaining` which is the syntax element indicating size information of a transformation coefficient, to a prefix bit string and a suffix bit string, by using a parameter (`cRiceParam`). An operation of binarizing `coeff_abs_level_remaining` which is the syntax element indicating size information of a transformation coefficient by using the binarizer 2210 will be described later.

The context modeler 2220 provides a probability model for encoding a bit string corresponding to a syntax element, to the regular coding engine 2232. In detail, the context modeler 2220 outputs a probability of a binary value for encoding each binary value of a bit string of a current syntax element, to the binary arithmetic coder 2230.

A context model is a probability model of a bin, and includes information about which of 0 and 1 corresponds to a most probable symbol (MPS) and a least probable symbol (LPS) and an MPS or LPS probability.

The regular coding engine 2232 performs binary arithmetic coding regarding a bit string corresponding to a syntax element based on information about the MPS and the LPS provided by the context modeler 2220 and the probability information of the MPS or the LPS.

A context model used in encoding `coeff_abs_level_remaining`, which is the syntax element indicating size information of a transformation coefficient, may be set in advance according to a bin index of a transformation coefficient.

FIG. 23 is a block diagram illustrating a structure of a binarization apparatus 2300 according to an embodiment of the present invention. The binarization apparatus 2300 of FIG. 23 corresponds to the binarizer 2210 of FIG. 22.

Referring to FIG. 23, the binarization apparatus 2300 includes a parameter determining unit 2310 and a bit string generating unit 2320.

The parameter determining unit 2310 compares a size of a previous transformation coefficient that is encoded before a current transformation coefficient, with a predetermined critical value obtained based on a previous parameter used in binarization of a previous transformation coefficient level syntax element indicating a size of the previous transformation coefficient to thereby determine whether to renew the previous parameter. Also, the parameter determining unit 2310 obtains a

parameter to be used in binarization of a transformation coefficient level syntax element indicating a size of the current transformation coefficient by renewing or maintaining the previous parameter according to a result of the determination.

In detail, when the size of the previous transformation coefficient is $cLastAbsCoeff$, and the previous parameter is $cLastRiceParam$, the parameter determining unit 2310 determines a parameter $cRiceParam$ that is to be used in binarization of the transformation coefficient level syntax element, $coeff_abs_level_remaining$, which indicates the size of the current transformation coefficient, based on the following algorithm.

```
10      cRiceParam=
Min(cLastRiceParam+(cLastAbsLevel>(3*(1<<cLastRiceParam)))?1:0),4)
```

This algorithm may be implemented by the following pseudo code.

```
{
  If cLastAbsCoeff > 3*(1<<cLastRiceParam)
15      cRiceParam= Min (cLastRiceParam+1, 4)
}
```

As described in the above algorithm, the parameter determining unit 2310 compares a critical value th that is obtained based on the following equation:

```
20  th=3*(1<<cLastRiceParam), and cLastRiceParam. The parameter determining unit
2310 renews the previous parameter (cLastRiceParam) by increasing the same by 1
when cLastAbsCoeff is greater than the critical value  $th$ , and maintains the previous
parameter without renewing when cLastAbsCoeff is not greater than the critical value
 $th$ .
```

```
25      An initial parameter is set to 0. When a parameter (cRiceParam) is renewed,
it is gradually increased by +1 compared to the previous parameter
(cLastRiceParam). Also, the critical value  $th$  used in determining conditions for
renewal of a parameter is determined according to the parameter (cRiceParam), and
thus, as the parameter (cRiceParam) is renewed, the critical value  $th$  is also
30  gradually increased. That is, the critical value  $th$  is set to have a value that is
proportional to the previous parameter (cLastRiceParam), and when the previous
parameter (cLastRiceParam) is renewed, the parameter (cRiceParam) has a value
that is gradually increased by +1 compared to the previous parameter
(cLastRiceParam). As the parameter (cRiceParam) is renewed within the range of
```

0 to 4, the critical value th is gradually increased to 3, 6, 12, and 24.

The bit string generating unit 2320 binarizes the transformation coefficient level syntax element (`coeff_abs_level_remaining`) of a transformation coefficient by using a parameter, and outputs a bit string corresponding to the transformation coefficient level syntax element (`coeff_abs_level_remaining`) of a transformation coefficient.

In detail, the bit string generating unit 2320 obtains a parameter `cTrMax` according to the following equation: $cTrMax = 4 \ll cRiceParam$, by using the obtained parameter (`cRiceParam`). The parameter `cTrMax` is used as a standard for dividing the transformation coefficient level syntax element (`coeff_abs_level_remaining`) into a prefix and a suffix.

The bit string generating unit 2320 divides a value of the transformation coefficient level syntax element (`coeff_abs_level_remaining`) based on the parameter `cTrMax` so as to obtain a prefix having a value not exceeding the parameter `cTrMax` and a suffix indicating a portion that exceeds the parameter `cTrMax`. The bit string generating unit 2320 determines a prefix within a range not exceeding `cTrMax` according to the following equation: $Prefix = \text{Min}(cTrMax, \text{coeff_abs_level_remaining})$. A suffix exists only when the transformation coefficient level syntax element (`coeff_abs_level_remaining`) has a greater value than `cTrMax`. The suffix is a value corresponding to $(\text{coeff_abs_level_remaining} - cTrMax)$. When the transformation coefficient level syntax element (`coeff_abs_level_remaining`) does not exceed `cTrMax`, only a prefix exists. For example, when the transformation coefficient level syntax element (`coeff_abs_level_remaining`) is 10, and the parameter `cTrMax` is 7, the transformation coefficient level syntax element (`coeff_abs_level_remaining`) is sorted into a prefix having a value of 7 and a suffix having a value of 3. Alternatively, when the transformation coefficient level syntax element (`coeff_abs_level_remaining`) is 6, and the parameter `cTrMax` is 7, the transformation coefficient level syntax element (`coeff_abs_level_remaining`) is sorted into a prefix having a value of 6 and does not have a suffix.

When a prefix and a suffix are determined by dividing the value of the transformation coefficient level syntax element (`coeff_abs_level_remaining`) based on the parameter `cTrMax`, the bit string generating unit 2320 binarizes the prefix and the suffix by using a predetermined binarization method that is set in advance, to output bit strings corresponding to the prefix and the suffix. For example, the bit

string generating unit 2320 may output a bit string by binarizing a prefix having a value corresponding to $\text{Min}(\text{cTrMax}, \text{coeff_abs_level_remaining})$ by using a truncated unary binarization method, and may output a bit string by binarizing a suffix having a value corresponding to $(\text{coeff_abs_level_remaining} - \text{cTrMax})$ by using a k-th exponential Golomb method. A value of k may be determined by using the parameter (cRiceParam) determined by using the parameter determining unit 2310. For example, the value of k may have a value of $\text{cRiceParam} + 1$.

According to the truncated unary binarization method, the prefix having a value of $\text{Min}(\text{cTrMax}, \text{coeff_abs_level_remaining})$ may be binarized as shown in Table 2 below.

[Table 2]

Min (cTrMax, coeff_abs_level_remaining)	Output bit string
0	0
1	10
2	110
3	1110
.....	

The bit string generating unit 2320 may generate a bit string corresponding to a prefix and a suffix according to the parameter (cRiceParam) and by referring to a table set in advance. According to a lookup table method, the table set in advance may be set such that a length of a bit string corresponding to a relatively great value is reduced as the value of the parameter (cRiceParam) increases.

An operation of entropy encoding a syntax element related to a transformation unit according to an embodiment of the present invention will be described in detail with reference to FIGS. 15 through 21.

FIG. 15 illustrates a transformation unit 1500 that is entropy encoded according to an embodiment of the present invention. While the transformation unit 1500 having a 16x16 size is illustrated in FIG. 15, the size of the transformation unit 1500 is not limited to the illustrated size of 16x16 but may also be of various sizes ranging from 4x4 to 32x32.

Referring to FIG. 15, for entropy encoding and decoding of the transformation coefficient included in the transformation unit 1500, the transformation unit 1500 may be divided into smaller transformation units. Hereinafter, an operation of entropy encoding a syntax element related to a 4x4 transformation unit 1510 included in the

transformation unit 1500 will be described. This operation of entropy encoding a syntax element related to the 4x4 transformation unit 1510 may also be applied to a transformation unit of different sizes.

Transformation coefficients included in the 4x4 transformation unit 1510 each have a transformation coefficient (absCoeff) as illustrated in FIG. 15. The transformation coefficients included in the 4x4 transformation unit 1510 may be serialized according to a predetermined scanning order as illustrated in FIG. 15 and sequentially processed. However, the scanning order is not limited as illustrated but may also be modified.

As described above, examples of syntax elements related to the 4x4 transformation unit 1510 are significant_coeff_flag which is a syntax indicating whether each transformation coefficient included in a transformation unit is a significant transformation coefficient having a value that is not 0, coeff_abs_level_greater1_flag which is a syntax element indicating whether an absolute value of the transformation coefficient is greater than 1, coeff_abs_level_greater2_flag which is a syntax element indicating whether the absolute value is greater than 2, and coeff_abs_level_remaining which is a syntax element indicating size information of the remaining transformation coefficients.

FIG. 16 illustrates a significance map SigMap 1600 corresponding to the transformation unit 1510 of FIG. 15 according to an embodiment of the present invention.

Referring to FIGS. 15 and 16, the significance map SigMap 1600 having a value of 1 for each of the significant transformation coefficients that have a value that is not 0, from among the transformation coefficients included in the 4x4 transformation unit 1510 of FIG. 15, is set. The significance map SigMap 1600 is entropy encoded or decoded by using a previously set context model.

FIG. 17 illustrates coeff_abs_level_greater1_flag 1700 corresponding to the 4x4 transformation unit 1510 of FIG. 15.

Referring to FIGS. 15 through 17, coeff_abs_level_greater1_flag 1700 which is a flag indicating whether a corresponding significance transformation coefficient has a value greater than 1, regarding significant transformation coefficients for which the significance map SigMap 1600 has a value of 1, is set. When coeff_abs_level_greater1_flag 1700 is 1, it indicates that a corresponding transformation coefficient is a transformation coefficient having a value greater than

1, and when `coeff_abs_level_greater1_flag` 1700 is 0, it indicates that a corresponding transformation coefficient is a transformation coefficient having a value of 1. In FIG. 17, when `coeff_abs_level_greater1_flag` 1710 is at a position of a transformation coefficient having a value of 1, the `coeff_abs_level_greater1_flag` 1710 has a value of 0.

FIG. 18 illustrates `coeff_abs_level_greater2_flag` 1800 corresponding to the 4x4 transformation unit 1510 of FIG. 15.

Referring to FIGS. 15 through 18, `coeff_abs_level_greater2_flag` 1800 indicating whether a corresponding transformation coefficient has a value greater than 2, regarding transformation coefficients for which the `coeff_abs_level_greater1_flag` 1700 is set to 1, is set. When `coeff_abs_level_greater2_flag` 1800 is 1, it indicates that a corresponding transformation coefficient is a transformation coefficient having a value greater than 2, and when `coeff_abs_level_greater2_flag` 1800 is 0, it indicates that a corresponding transformation coefficient is a transformation coefficient having a value of 2. In FIG. 18, when `coeff_abs_level_greater2_flag` 1810 is at a position of a transformation coefficient having a value of 2, the `coeff_abs_level_greater2_flag` 1810 has a value of 0.

FIG. 19 illustrates `coeff_abs_level_remaining` 1910 corresponding to the 4x4 transformation unit 1510 of FIG. 15.

Referring to FIGS. 15 through 19, `coeff_abs_level_remaining` 1900 which is a syntax element indicating size information of the remaining transformation coefficients may be obtained by calculating $(absCoeff - baseLevel)$ of each transformation coefficient.

As described above, `coeff_abs_level_remaining` 1900 which is the syntax element indicating size information of the remaining transformation coefficients has a difference in a range between the size of the transformation coefficient (`absCoeff`) and a basic level value `baseLevel` determined by using `coeff_abs_level_greater1_flag` and `coeff_abs_level_greater2_flag`. The basic level value `baseLevel` is determined according to the equation:
 $baseLevel = 1 + coeff_abs_level_greather1_flag + coeff_abs_level_greather2_flag$, and `coeff_abs_level_remaining` is determined according to the equation:
 $coeff_abs_level_remaining = absCoeff - baseLevel$.

The parameter determining unit 2310 reads the `coeff_abs_level_remaining`

1900 according to the illustrated scanning order to obtain sizes of transformation coefficients such as '0 3 12 3 3 3 4 4 5 5 8 8'.

The parameter determining unit 2310 sequentially determines a parameter (cRiceParam) used in binarization of size information of each transformation coefficient according to a scanning order. First, an initial parameter (cRiceParam) is set to 0. According to the above-described algorithm, the parameter increases only when a condition $cLastAbsCoeff > 3^{*(1 \leq cLastRiceParam)}$ is satisfied. The initially set parameter (cRiceParam) is 0, and it maintains this value until a size of a previous transformation coefficient (cLastAbsCoeff) has a value of $3^{*(1 \leq 0)}$, that is, a value greater than 3. Referring to FIG. 19, a size of a third transformation coefficient, '12' (1920), is greater than 3, and thus, when a size of a transformation coefficient, which comes after the transformation coefficient '12' (1920), is binarized, a parameter (cRiceParam) having a value renewed from 0 to 1 is used. When the parameter (cRiceParam) is renewed to 1, the parameter (cRiceParam) is renewed again only when a condition of $cLastAbsCoeff > 3^{*(1 \leq 1)}$, that is, $cLastAbsCoeff > 6$, is satisfied. Referring to FIG. 19, '8' (1930) which is a size of a second transformation coefficient from the last is greater than 6, and thus, the parameter (cRiceParam) is renewed from 1 to 2.

FIG. 20 illustrates a table showing syntax elements related to the transformation units 1510, 1600, 1700, 1800, and 1900 illustrated in FIGS. 15 through 19. In FIG. 20, GTR1 denotes coeff_abs_level_greater1_flag; GTR2 denotes coeff_abs_level_greater2_flag, and Remaining denotes coeff_abs_level_remaining. Referring to FIG. 20, a syntax element indicating a transformation coefficient level, coeff_abs_level_remaining, is not a binary value and thus is binarized by using a parameter.

FIG. 21 illustrates another example of coeff_abs_level_remaining that is binarized according to an embodiment of the present invention.

As described above, an initial parameter (cRiceParam) is set to 0, and increases by +1 only when a condition of $cLastAbsCoeff > 3^{*(1 \leq cLastRiceParam)}$ is satisfied. The initial parameter (cRiceParam) has a value of Min (cLastRiceParam+1, 4), and thus, the renewed parameter may not have a value exceeding 4. A critical value, $3^{*(1 \leq cLastRiceParam)}$, that is used in determining whether to renew a parameter has a value of $3^{*(1 \leq 0)}$, $3^{*(1 \leq 1)}$, $3^{*(1 \leq 2)}$ or $3^{*(1 \leq 3)}$ according to a previous parameter (cRiceParam) used in the binarization of

the size of the previous transformation coefficient. Accordingly, the parameter (cRiceParam) increases by +1 after processing of a transformation coefficient having a value greater than 3, and then, increases by +1 after processing of a transformation coefficient that is greater than 6, and then increases by +1 after processing of a transformation coefficient that is greater than 12, and increases by +1 a final time after a transformation coefficient greater than 24 is processed. That is, the parameter (cRiceParam) gradually increases by +1 also when abrupt variation in values exist from among transformation coefficients.

Referring to FIG. 21, the parameter (cRiceParam) having the initially set value of 0 increases by +1 after processing of a transformation coefficient 2110 that has a value of 12 and is greater than 3 for the first time. After the transformation unit 2110 having a value of 12, the renewed parameter (cRiceParam) is maintained until a transformation unit greater than 6 which is a next critical value is processed. After a transformation coefficient 2120 having a value of 8 that is greater than 6, which is the next critical value, is processed, the parameter (cRiceParam) increases by +1 to have a value of 2. After the transformation unit 2120 having a value of 8, the renewed parameter (cRiceParam) is maintained until a transformation unit greater than 12 which is a next critical value is processed. After the transformation unit 2120 having a value of 8, after a transformation coefficient 2130 having a value of 13 that is greater than 12, which is the next critical value, is processed, the parameter (cRiceParam) increases by +1 to have a value of 3. After the transformation unit 2130 having a value of 13, the renewed parameter (cRiceParam) is maintained until a transformation unit greater than 24 which is a next critical value is processed. After the transformation unit 2130 having a value of 13, after a transformation coefficient 2140 having a value of 25 that is greater than 24, which is the next critical value, is processed, the parameter (cRiceParam) increases by +1 to have a value of 4. In binarization operations of transformation coefficients after the transformation coefficient 2140 having a value of 25, since the parameter (cRiceParam) has reached the maximum of 4, the parameter (cRiceParam) of 4 is used, and the renewing operation is no longer performed.

As described above, when the parameter (cRiceParam) used in binarization of the transformation coefficient level syntax element `coeff_abs_level_remaining` indicating a size of a current transformation coefficient is determined by using the parameter determining unit 2310, the bit string generating unit 2320 sorts the

transformation coefficient level syntax element `coeff_abs_level_remaining` into a prefix and a suffix based on the parameter (`cTrMax`) that is determined by using the parameter (`cRiceParam`), and binarizes the prefix and the suffix by applying binarization methods set in advance with respect to the prefix and the suffix to thereby output bit strings corresponding to `coeff_abs_level_remaining`.

FIG. 24 is a flowchart illustrating an entropy encoding method for syntax elements indicating a transformation coefficient level according to an embodiment of the present invention.

Referring to FIG. 24, in operation 2410, the parameter determining unit 2310 obtains transformation coefficient level syntax elements (`coeff_abs_level_remaining`) indicating sizes of transformation coefficients included in a transformation unit according to a predetermined scanning order.

In operation 2420, the parameter determining unit 2310 compares a size of a previous transformation coefficient (`cLastAbsCoeff`) that is encoded before a current transformation coefficient, with a predetermined critical value that is obtained based on the previous parameter (`cLastRiceParam`) used in binarization of previous transformation coefficient level syntax elements indicating the size of the previous transformation coefficient (`cLastAbsCoeff`), to thereby determine whether to renew the previous parameter (`cLastRiceParam`).

In operation 2430, the parameter determining unit 2310 renews or maintains the previous parameter based on a result of determination of operation 2420 to thereby obtain a parameter used in binarization of a transformation coefficient level syntax element indicating a size of the current transformation coefficient. As described above, the parameter determining unit 2310 compares a critical value $th = 3 * (1 < cLastRiceParam)$ with the previous parameter `cLastRiceParam`; and when `cLastAbsCoeff` is greater than th , the parameter determining unit 2310 renews the previous parameter by increasing the same by 1; and when `cLastAbsCoeff` is not greater than th , the parameter determining unit 2310 does not renew but maintains the previous parameter. When the previous parameter is renewed, the renewed parameter is gradually increased by +1.

In operation 2440, the bit string generating unit 2320 binarizes the transformation coefficient level syntax element (`coeff_abs_level_remaining`) by using the obtained parameter (`cRiceParam`), thereby outputting a bit string corresponding to the transformation coefficient level syntax element (`coeff_abs_level_remaining`) of

the current transformation coefficient.

According to the operation of entropy encoding a transformation coefficient level syntax element according to the embodiments of the present invention described above, even when there is a transformation coefficient having an abruptly large value from among transformation coefficients that are processed according to a predetermined scanning order, a parameter may not have to be abruptly modified but may gradually increase by +1.

Meanwhile, an operation of updating a parameter in order to entropy encode a transformation coefficient level syntax element according to the embodiments of the present invention described above may also be applied in the binarization of syntax elements other than the transformation coefficient level syntax element.

The operation of updating a parameter according to an embodiment of the present invention may be applied in updating a parameter used in binarization of other syntax elements by using a Golomb-Rice code. In addition, the method of updating a parameter according to an embodiment of the present invention may be applied in updating a parameter used in binarization of a syntax element by applying a binarization method such as a concatenate code binarization method. When a concatenate code is used, syntax elements are sorted into a prefix and a suffix, and the method of updating a parameter according to an embodiment of the present invention may be applied in updating a predetermined parameter for the purpose of determining a prefix and a suffix. Similarly, the method of updating a parameter according to an embodiment of the present invention may be applied in updating a parameter used in encoding a syntax element by using a fixed length code and a variable length code (VLC) table as in a low complexity entropy coding (LCEC) method.

FIG. 25 is a block diagram illustrating an entropy decoding apparatus 2500 according to an embodiment of the present invention. The entropy decoding apparatus 2500 corresponds to the entropy decoder 220 of the video decoding apparatus 200 of FIG. 2. The entropy decoding apparatus 2500 performs a reverse operation of the entropy encoding operation performed by the entropy encoding apparatus 2200 described above.

Referring to FIG. 25, the entropy decoding apparatus 2500 includes a context modeler 2510, a regular decoding engine 2520, a bypass decoding engine 2530, and a de-binarizer 2540.

A syntax element encoded by using bypass coding is output to the bypass decoder 2530 to be decoded, and a syntax element encoded by regular coding is decoded by using the regular decoder 2520. The regular decoder 2520 arithmetically decodes a binary value of a current syntax element based on a context model provided by using the context modeler 2510 to thereby output a bit string. A context model used in arithmetic decoding of a syntax element indicating size information of a transformation coefficient, `coeff_abs_level_remaining`, may be set in advance according to a bin index of a transformation coefficient.

The de-binarizer 2540 restores bit strings that are arithmetically decoded by using the regular decoding engine 2520 or the bypass decoding engine 2530, to syntax elements again.

The entropy decoding apparatus 2500 arithmetically decodes syntax elements related to transformation units, such as `SigMap`, `coeff_abs_level_greater1_flag`, or `coeff_abs_level_greater2_flag`, in addition to `coeff_abs_level_remaining`, and outputs the same. When the syntax elements related to a transformation unit are restored, data included in the transformation units may be decoded by using inverse quantization, reverse transformation, and predictive decoding, based on the restored syntax elements.

FIG. 26 is a block diagram illustrating a structure of a de-binarization apparatus 2600 according to an embodiment of the present invention. The de-binarization apparatus 2600 of FIG. 26 corresponds to the de-binarizer 2540 of FIG. 25.

Referring to FIG. 26, the de-binarization apparatus 2600 includes a parameter determining unit 2610 and a syntax element restoring unit 2620.

The parameter determining unit 2610 compares a size of a previous transformation coefficient that is decoded before a current transformation coefficient, with a predetermined critical value that is obtained based on the previous parameter used in de-binarization of a previous transformation coefficient level syntax element indicating a size of the previous transformation coefficient, to thereby determine whether to renew the previous parameter. The parameter determining unit 2610 renews or maintains the previous parameter based on a result of the determination to thereby obtain a parameter used in de-binarization of a transformation coefficient level syntax element indicating a size of the current transformation coefficient. In the same manner as the parameter determining unit 2310 of FIG. 23 described

above, the parameter determining unit 2610 compares a critical value th obtained based on the equation: $th=3*(1<cLastRiceParam)$ with a previous parameter $cLastRiceParam$. When $cLastAbsCoeff$ is greater than th , the parameter determining unit 2610 renews the previous parameter ($cLastRiceParam$) by increasing the same by 1; and when $cLastAbsCoeff$ is not greater than th , the parameter determining unit 2610 does not renew but maintains the previous parameter ($cLastRiceParam$).

The syntax element restoring unit 2620 de-binarizes a bit string corresponding to the current transformation coefficient level syntax element by using the obtained parameter to thereby restore a syntax element ($coeff_abs_level_remaining$) indicating a size of the current transformation coefficient. In detail, the syntax element restoring unit 2620 sorts bit strings into a prefix bit string corresponding to a bit string that is obtained by binarizing a value corresponding to $Min(cTrMax, coeff_abs_level_remaining)$ by using a truncated unary binarization method and a suffix bit string corresponding to a bit string that is obtained by binarizing a value corresponding to $(coeff_abs_level_remaining-cTrMax)$ by using a k -th (k is $cRiceParam+1$) exponential Golomb method, and restores the syntax element ($coeff_abs_level_remaining$) by de-binarizing the prefix bit string by using the truncated unary binarization method and the suffix bit string by using the k -th exponential Golomb method.

FIG. 27 is a flowchart illustrating a method of entropy decoding of a transformation coefficient level according to an embodiment of the present invention.

Referring to FIG. 27, in operation 2710, transformation coefficient level syntax elements indicating sizes of transformation coefficients included in a transformation unit are parsed from a bit stream. The parsed transformation coefficient level syntax elements are bit strings each consisting of 0s and 1s.

In operation 2720, the parameter determining unit 2610 compares a size of a previous transformation coefficient ($cLastAbsCoeff$) that is restored before a current transformation coefficient, with a predetermined critical value obtained based on a previous parameter ($cLastRiceParam$) used in de-binarization of a previous transformation coefficient level syntax element indicating a size of the previous transformation coefficient ($cLastAbsCoeff$) to thereby determine whether to renew the previous parameter ($cLastRiceParam$).

In operation 2730, the parameter determining unit 2610 renews or maintains

the previous parameter (cLastRiceParam) based on a result of determination to thereby obtain a parameter (cRiceParam) used in de-binarization of a transformation coefficient level syntax element (coeff_abs_level_remaining) indicating a size of the current transformation coefficient. As described above, the parameter determining unit 2610 compares a critical value th obtained based on the equation:

$th = 3 * (1 < cLastRiceParam)$ with the previous parameter cLastRiceParam. When cLastAbsCoeff is greater than th, the parameter determining unit 2610 renews the previous parameter (cLastRiceParam) by increasing the same by 1; and when cLastAbsCoeff is not greater than th, the parameter determining unit 2610 does not renew but maintains the previous parameter. When the parameter is renewed, the renewed parameter is gradually increased by +1.

In operation 2740, the syntax element restoring unit 2620 de-binarizes the current transformation coefficient level syntax element by using the obtained parameter to obtain size information of the current transformation coefficient. As described above, as $coeff_abs_level_remaining = absCoeff - baseLevel$, coeff_abs_level_greater1_flag and coeff_abs_level_greater2_flag are restored in addition to coeff_abs_level_remaining, and when a basic level value baseLevel is determined according to the equation: $baseLevel = 1 + coeff_abs_level_greater1_flag + coeff_abs_level_greater2_flag$, the size of the current transformation coefficient may be determined according to the equation: $absCoeff = coeff_abs_level_remaining + baseLevel$.

The invention can also be embodied as computer readable codes on a computer readable recording medium. The computer readable recording medium is any data storage device that can store data which can be thereafter read by a computer system. Examples of the computer readable recording medium include read-only memory (ROM), random-access memory (RAM), CD-ROMs, magnetic tapes, floppy disks, optical data storage devices, etc. The computer readable recording medium can also be distributed over network coupled computer systems so that the computer readable code is stored and executed in a distributed fashion.

While this invention has been particularly shown and described with reference to exemplary embodiments thereof, it will be understood by those of ordinary skill in the art that various changes in form and details may be made therein without departing from the spirit and scope of the invention as defined by the appended claims. Therefore, the scope of the invention is defined not by the detailed

description of the invention but by the appended claims, and all differences within the scope will be construed as being included in the present invention.

What is claimed is:

1. A method of decoding video, the method comprising:

obtaining bin string corresponding to current transformation coefficient level information by arithmetic decoding a bitstream;

determining a current binarization parameter, for determining a prefix bin string from the bin string, to be one of

(i) a previous binarization parameter and

(ii) an updated value of the previous binarization parameter obtained by adding n to the previous binarization parameter,

where n is integer, and n is equal to or greater than 1,

based on a comparison between a predetermined value that is proportional to the previous binarization parameter and

a value of a previous transformation coefficient;

obtaining the current transformation coefficient level information by de-binarizing binary values of the prefix bin string using the determined current binarization parameter;

determining the value of the current transformation coefficient using the current transformation coefficient level information; and

decoding data included in a transformation unit using inverse quantization and reverse transformation based on the value of the current transformation coefficient,

wherein, when the value of the previous transformation coefficient is equal to or smaller than the predetermined value,

the current binarization parameter is determined to be the previous binarization parameter,

when the value of the previous transformation coefficient is larger than the predetermined value,

the current binarization parameter is determined to be the updated value, and wherein the current binarization parameter has a value equal to or smaller than a predetermined threshold value.