



US 20150146792A1

(19) **United States**

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

(10) **Pub. No.: US 2015/0146792 A1**

(43) **Pub. Date: May 28, 2015**

(54) **METHOD AND APPARATUS FOR ENCODING VIDEO BY USING TRANSFORMATION INDEX, AND METHOD AND APPARATUS FOR DECODING VIDEO BY USING TRANSFORMATION INDEX**

(60) Provisional application No. 61/320,826, filed on Apr. 5, 2010.

(30) **Foreign Application Priority Data**

Oct. 5, 2010 (KR) ..... 10-2010-0096920

(71) Applicant: **SAMSUNG ELECTRONICS CO., LTD.**, Suwon-si (KR)

**Publication Classification**

(72) Inventors: **Jung-hye MIN**, Suwon-si (KR); **Woo-jin HAN**, Suwon-si (KR); **Tammy LEE**, Seoul (KR); **Il-koo KIM**, Osan-si (KR); **Min-su CHEON**, Suwon-si (KR)

(51) **Int. Cl.**  
**H04N 19/60** (2006.01)  
**H04N 19/44** (2006.01)

(73) Assignee: **SAMSUNG ELECTRONICS CO., LTD.**, Suwon-si (KR)

(52) **U.S. Cl.**  
CPC ..... **H04N 19/60** (2014.11); **H04N 19/44** (2014.11)

(21) Appl. No.: **14/611,679**

(22) Filed: **Feb. 2, 2015**

(57) **ABSTRACT**

**Related U.S. Application Data**

(63) Continuation of application No. 13/079,983, filed on Apr. 5, 2011, now Pat. No. 8,982,961.

Encoding and decoding a video using transformation index that indicates information that indicates a structure of a transformation unit transforming data of a current coding unit.

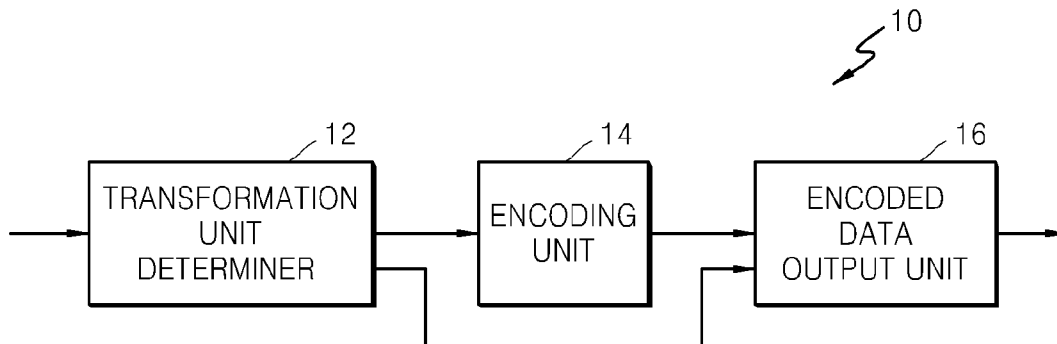


FIG. 1

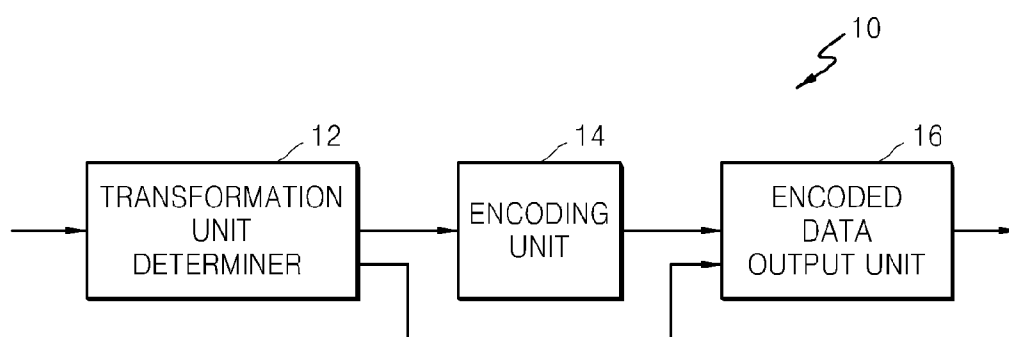


FIG. 2

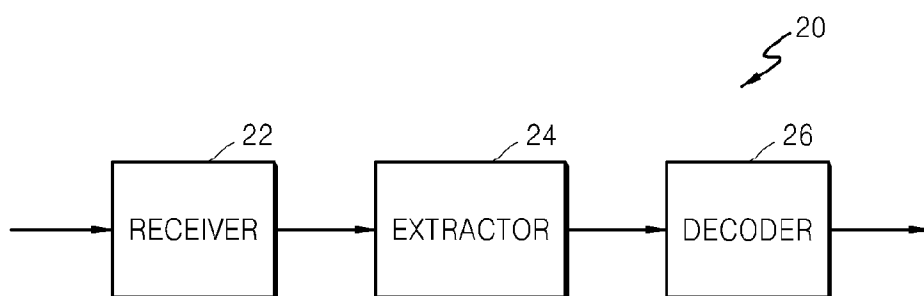


FIG. 3

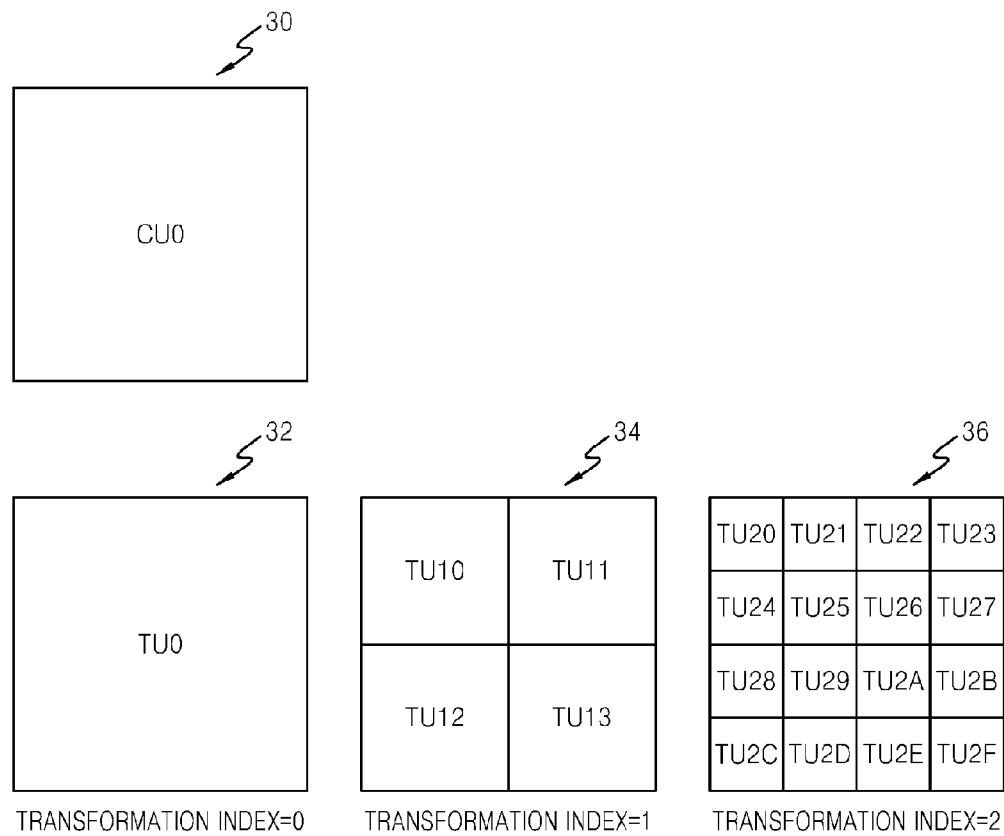
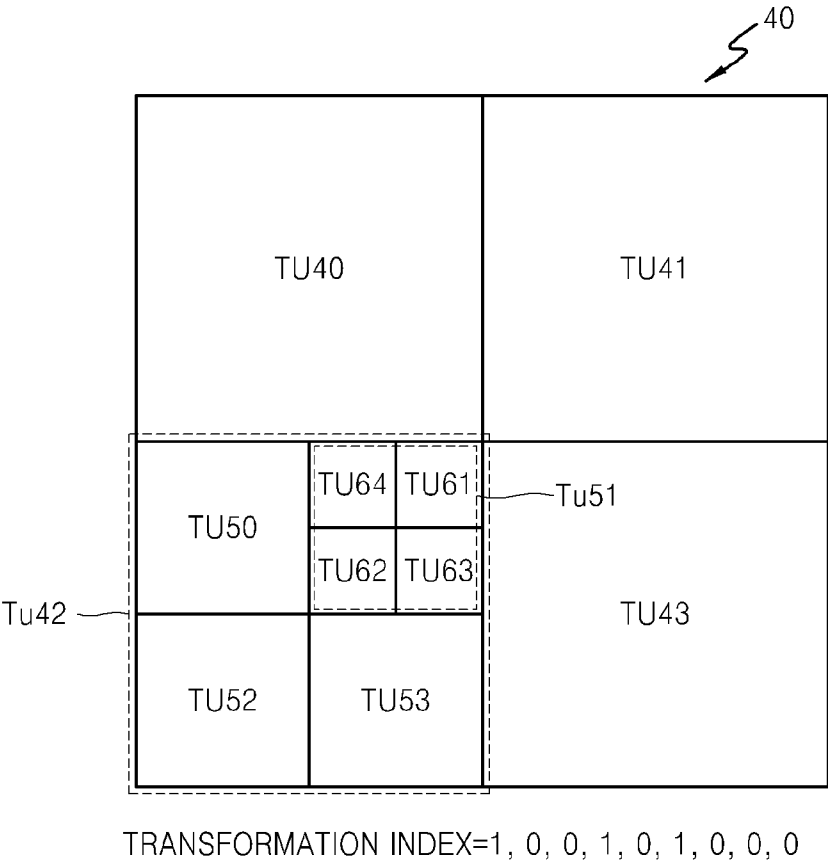
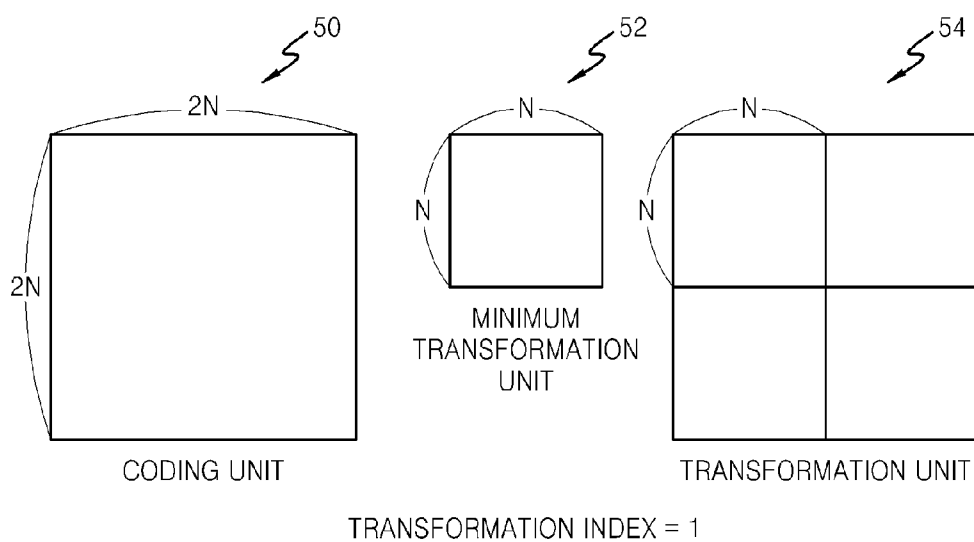


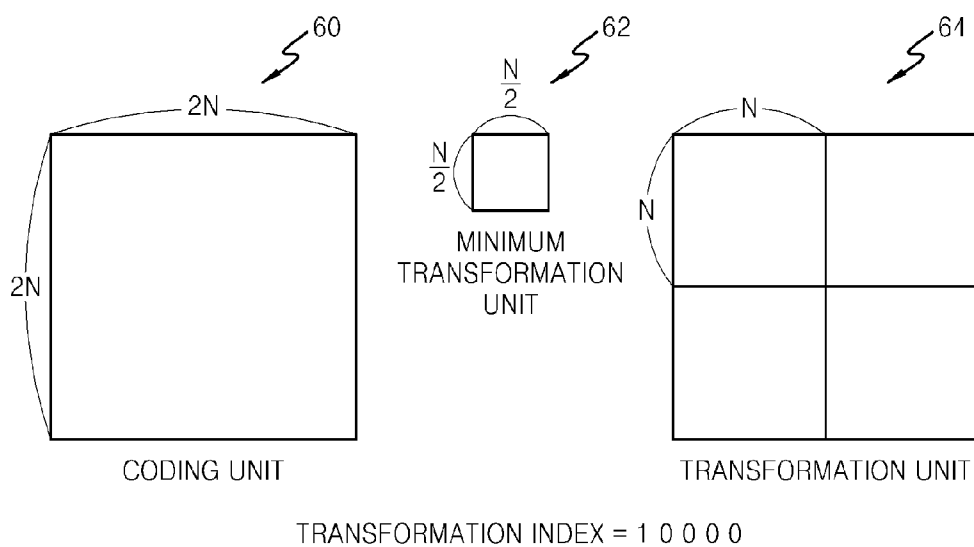
FIG. 4

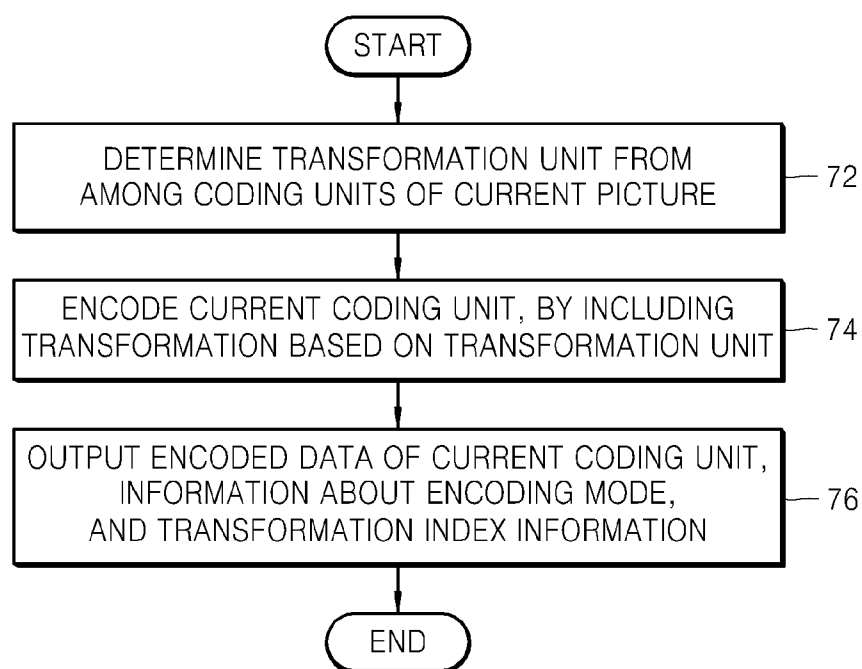
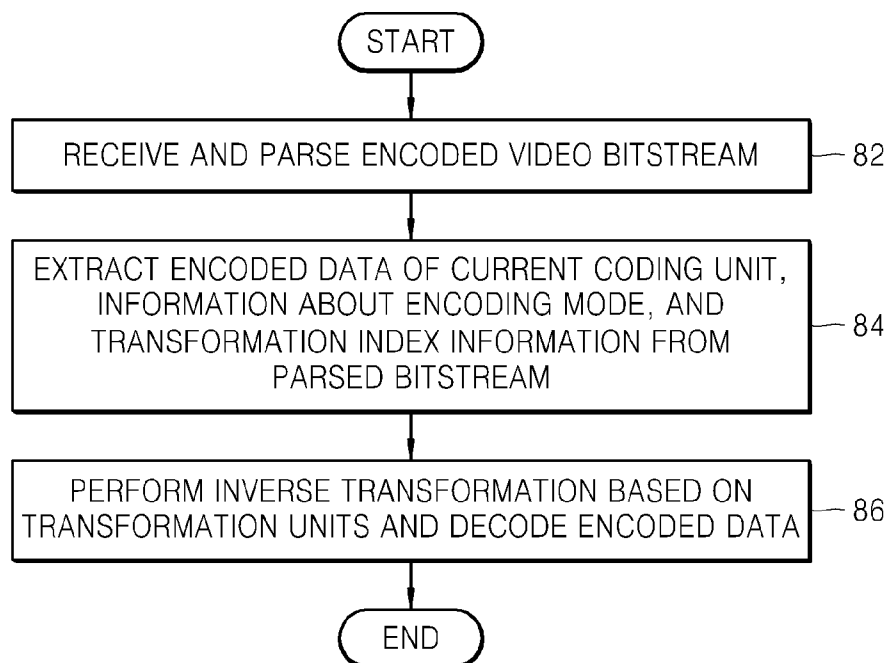


**FIG. 5**



**FIG. 6**



**FIG. 7****FIG. 8**

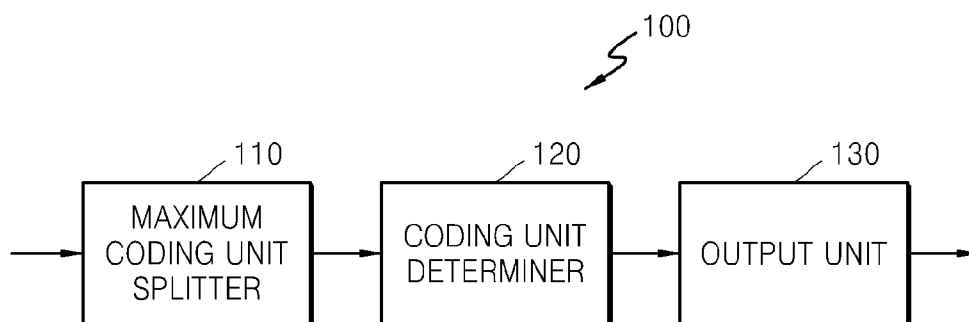
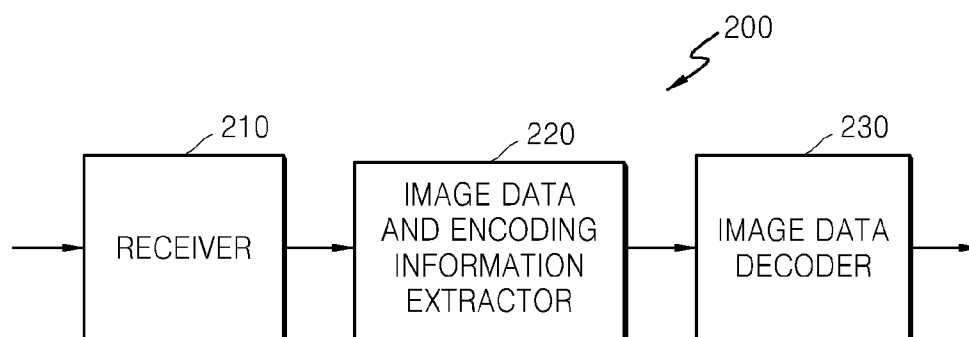
**FIG. 9****FIG. 10**

FIG. 11

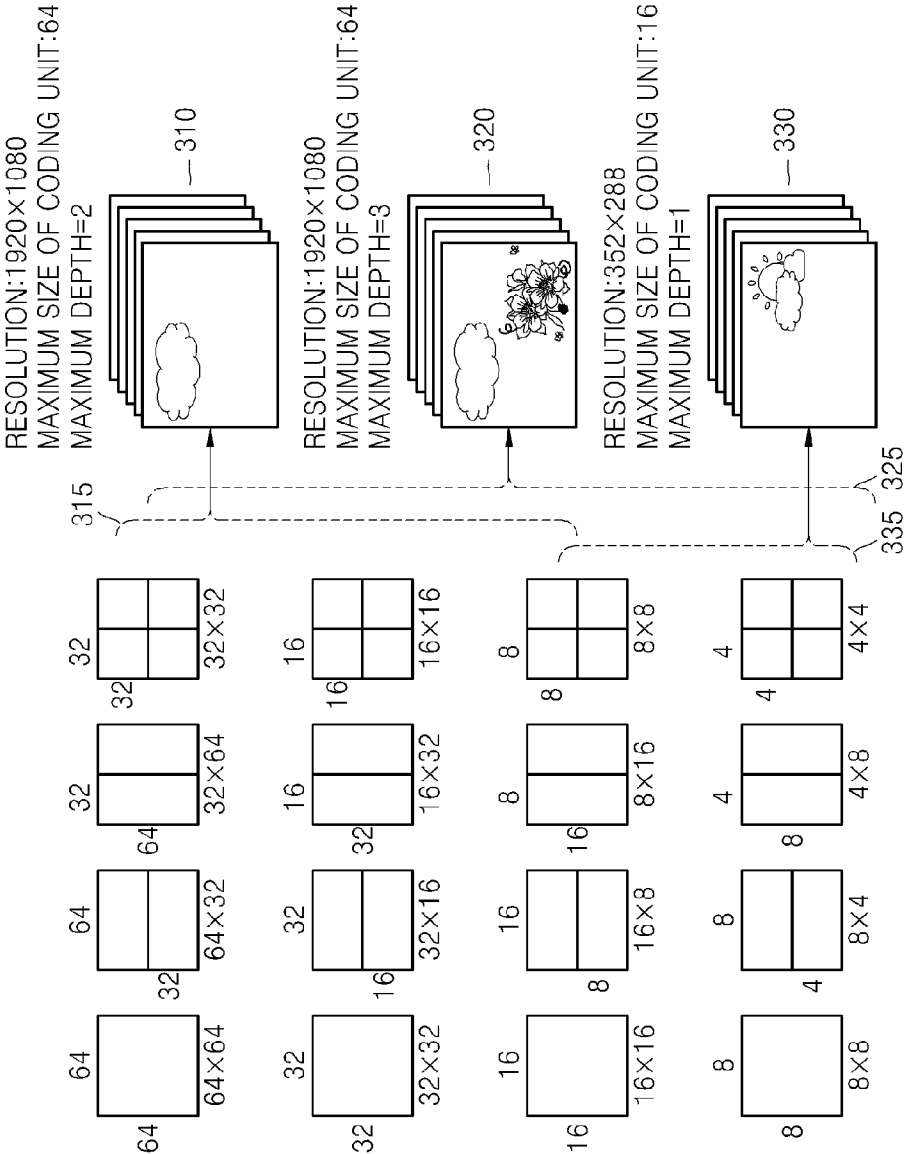




FIG. 12

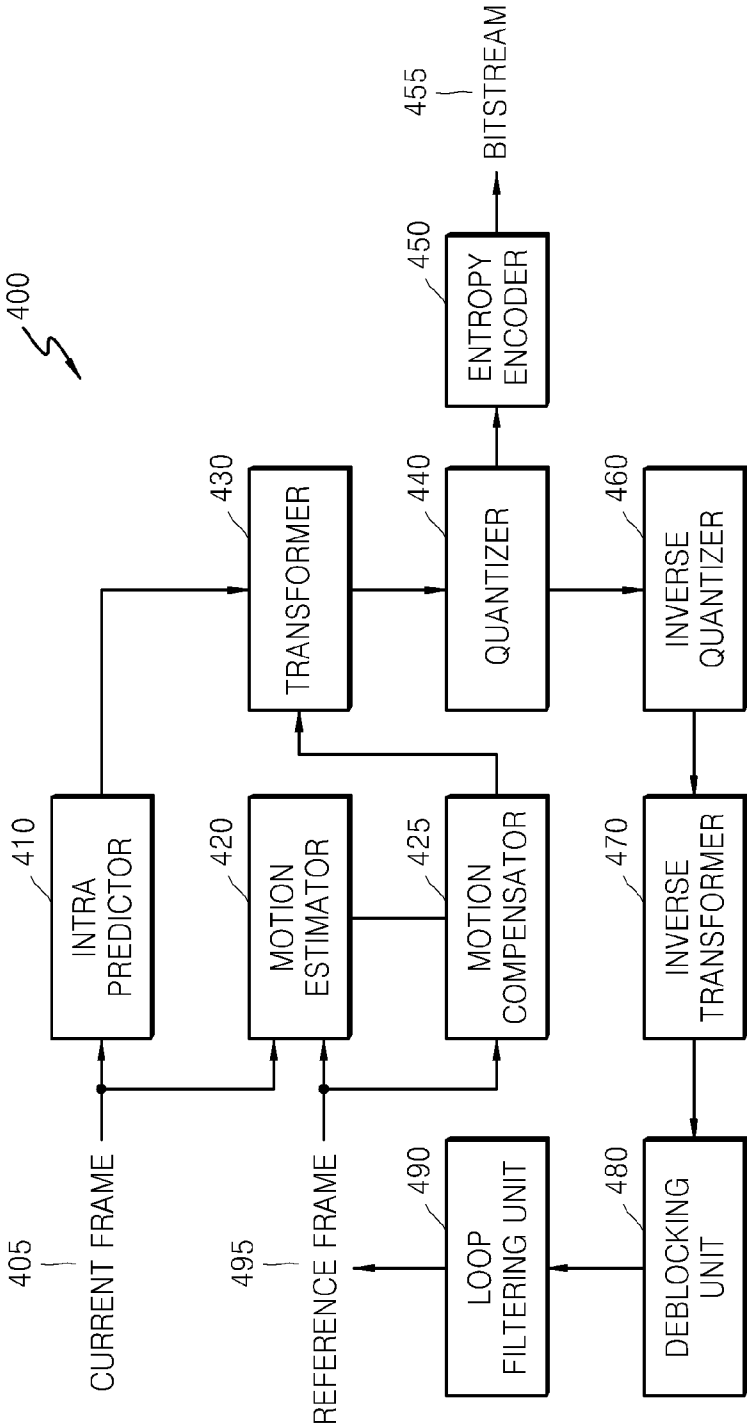


FIG. 13

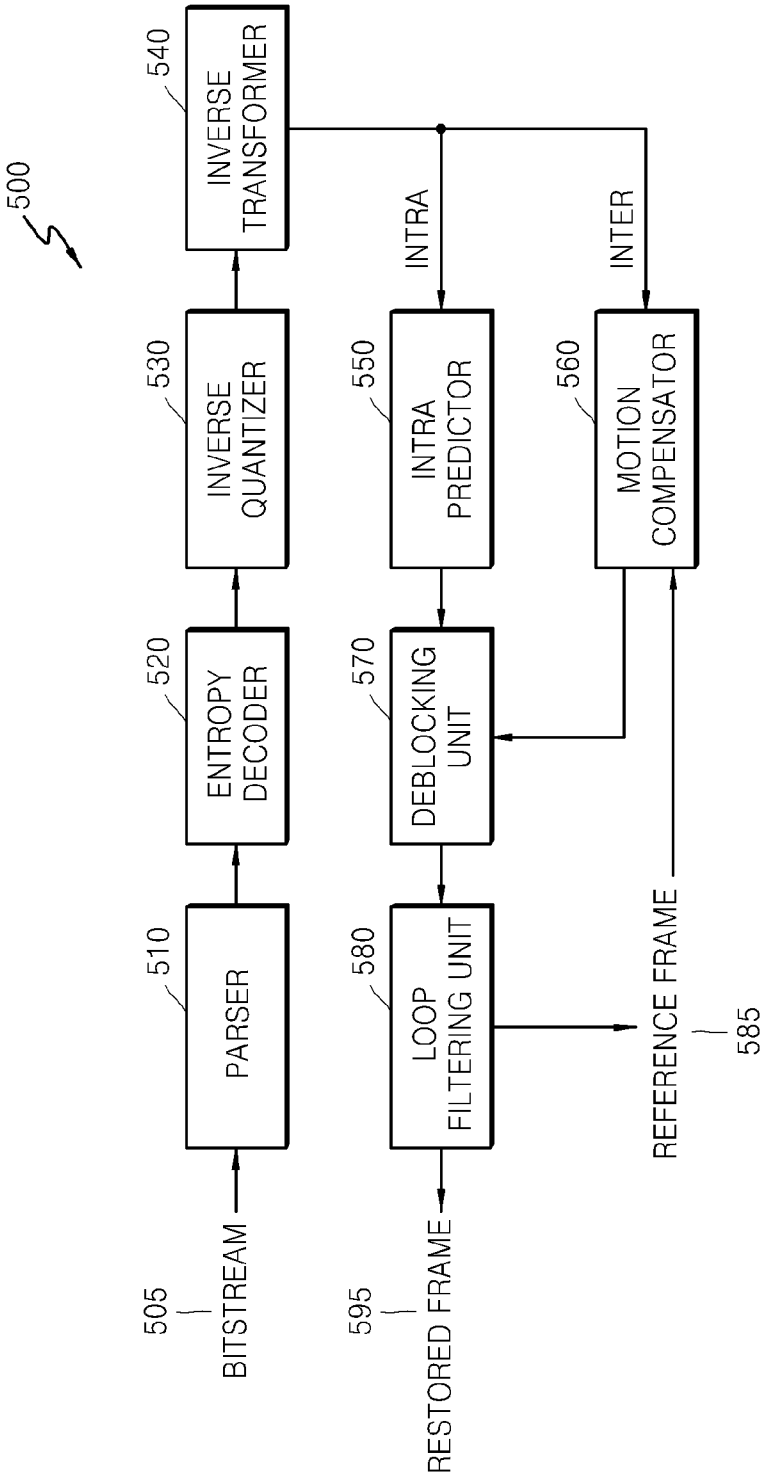


FIG. 14

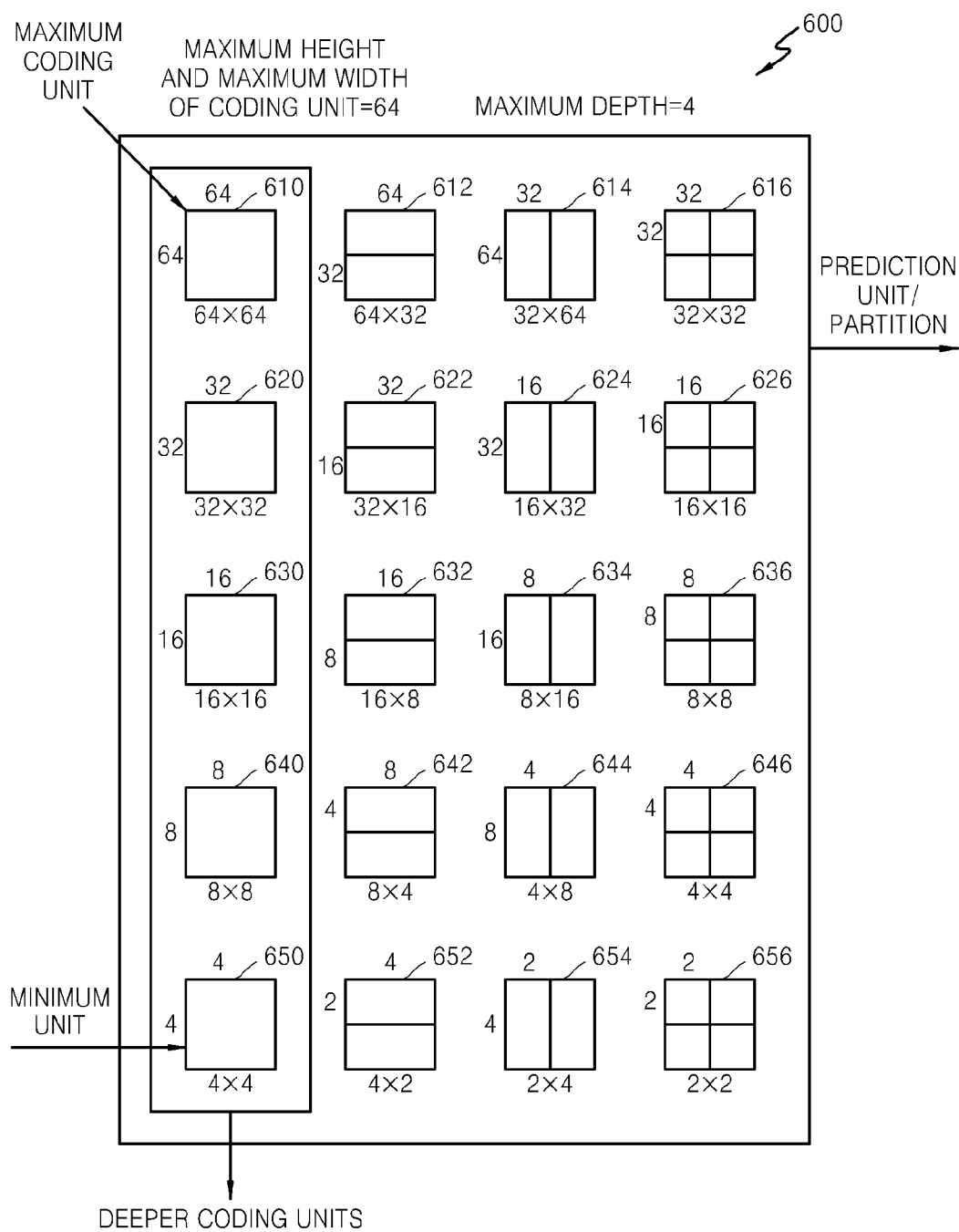


FIG. 15

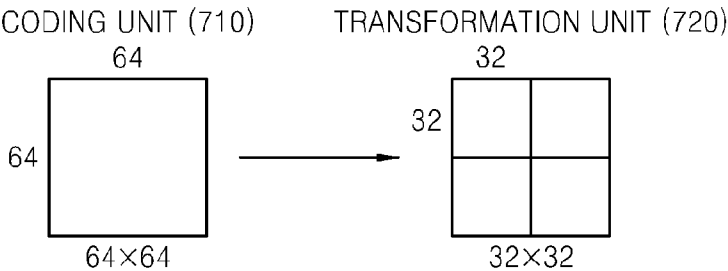


FIG. 16

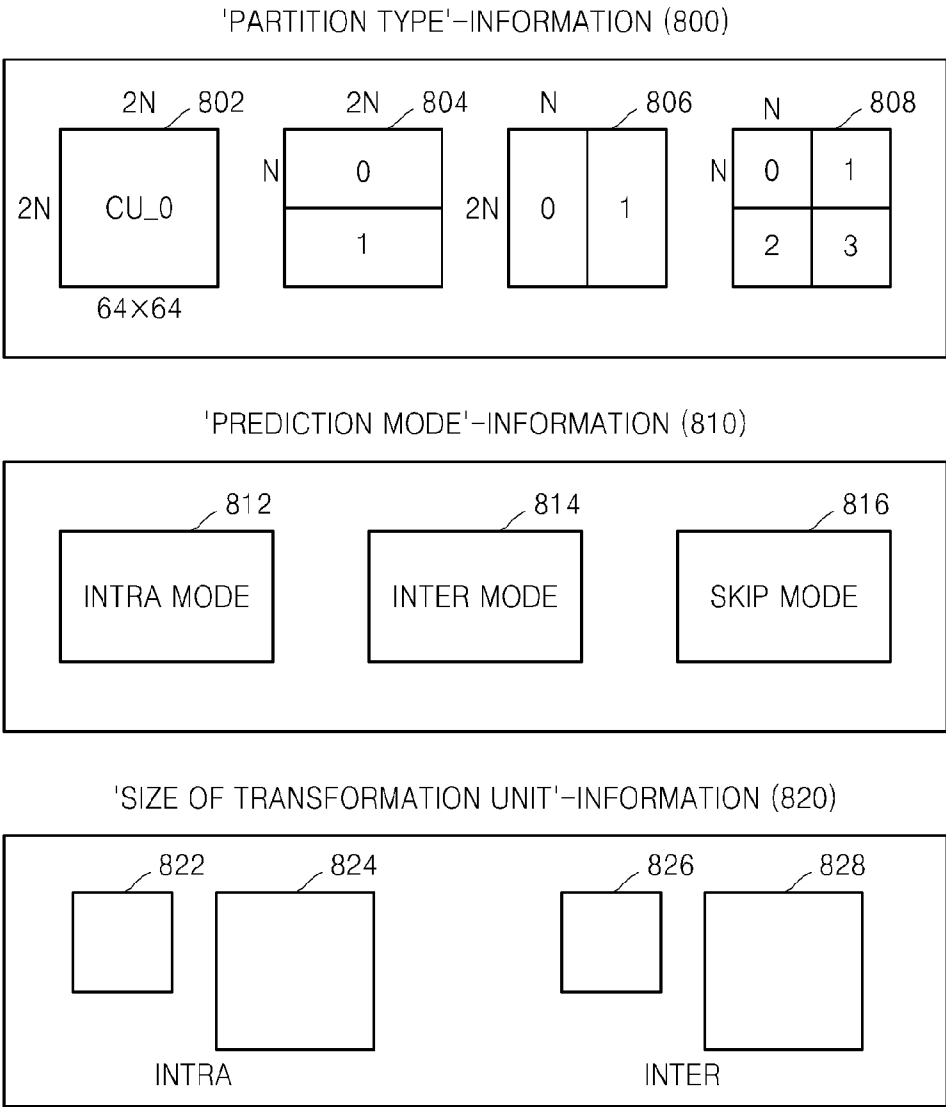
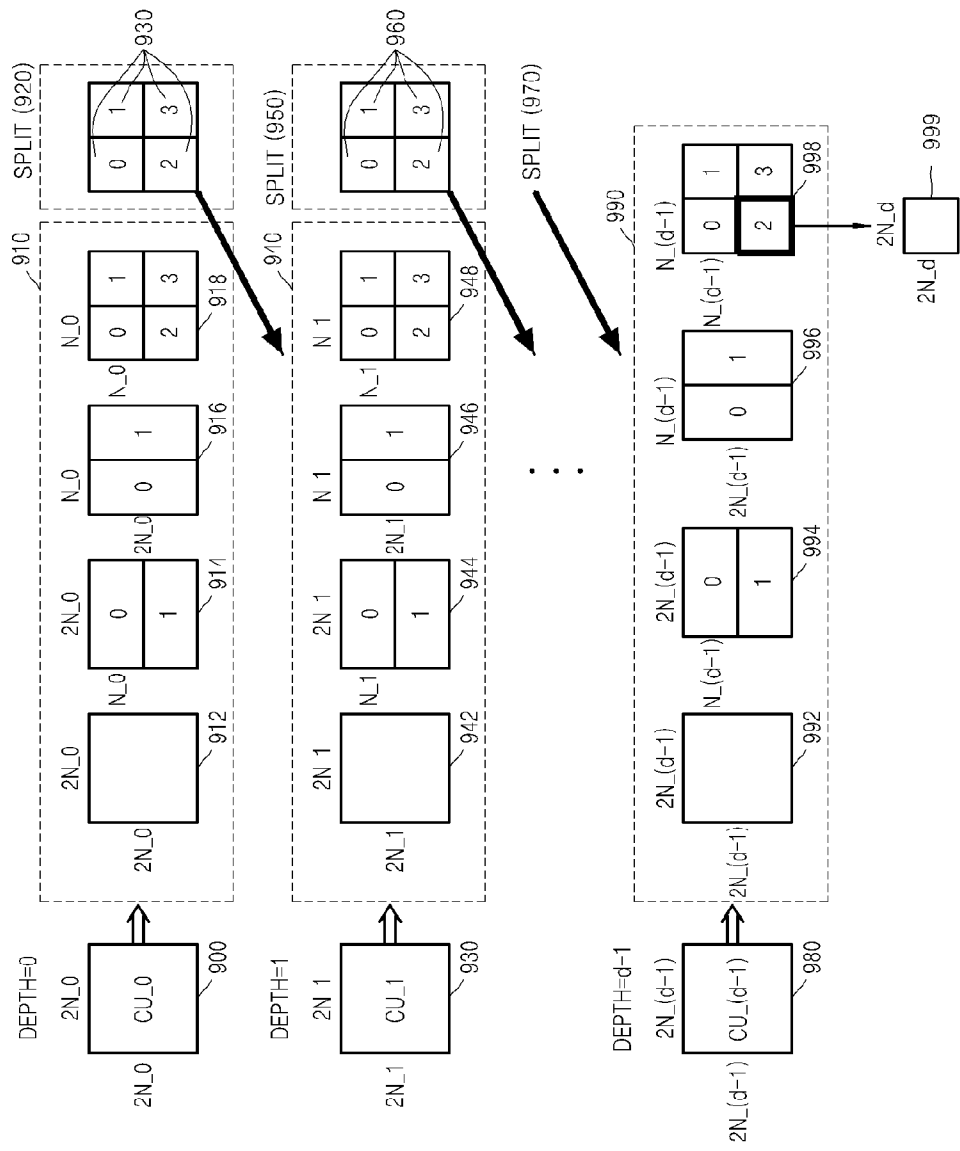


FIG. 17



1012				1014		1016		
				1018		1020	1022	
						1024	1026	
1028	1030		1032		1054			
	1040	1042	1048					
	1044	1046						
1050	1052							

## CODING UNITS (1010)

**FIG. 19**

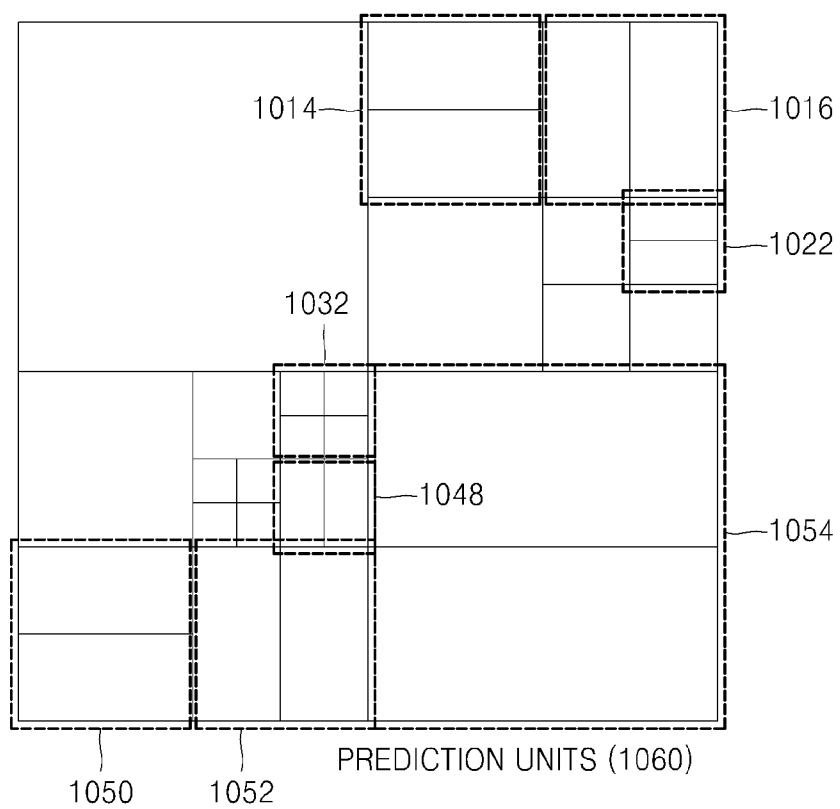


FIG. 20

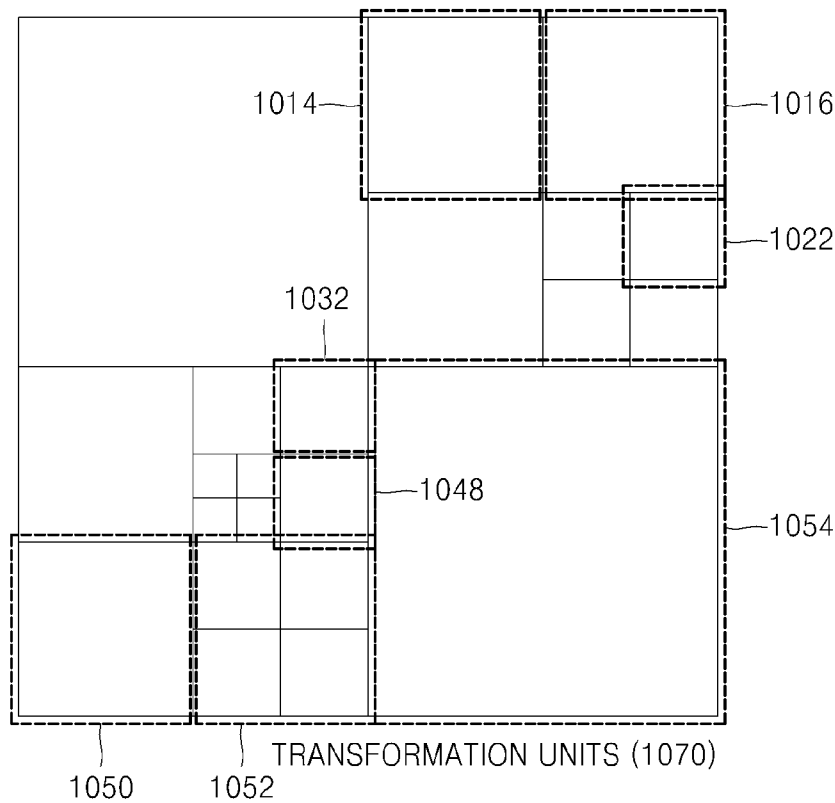
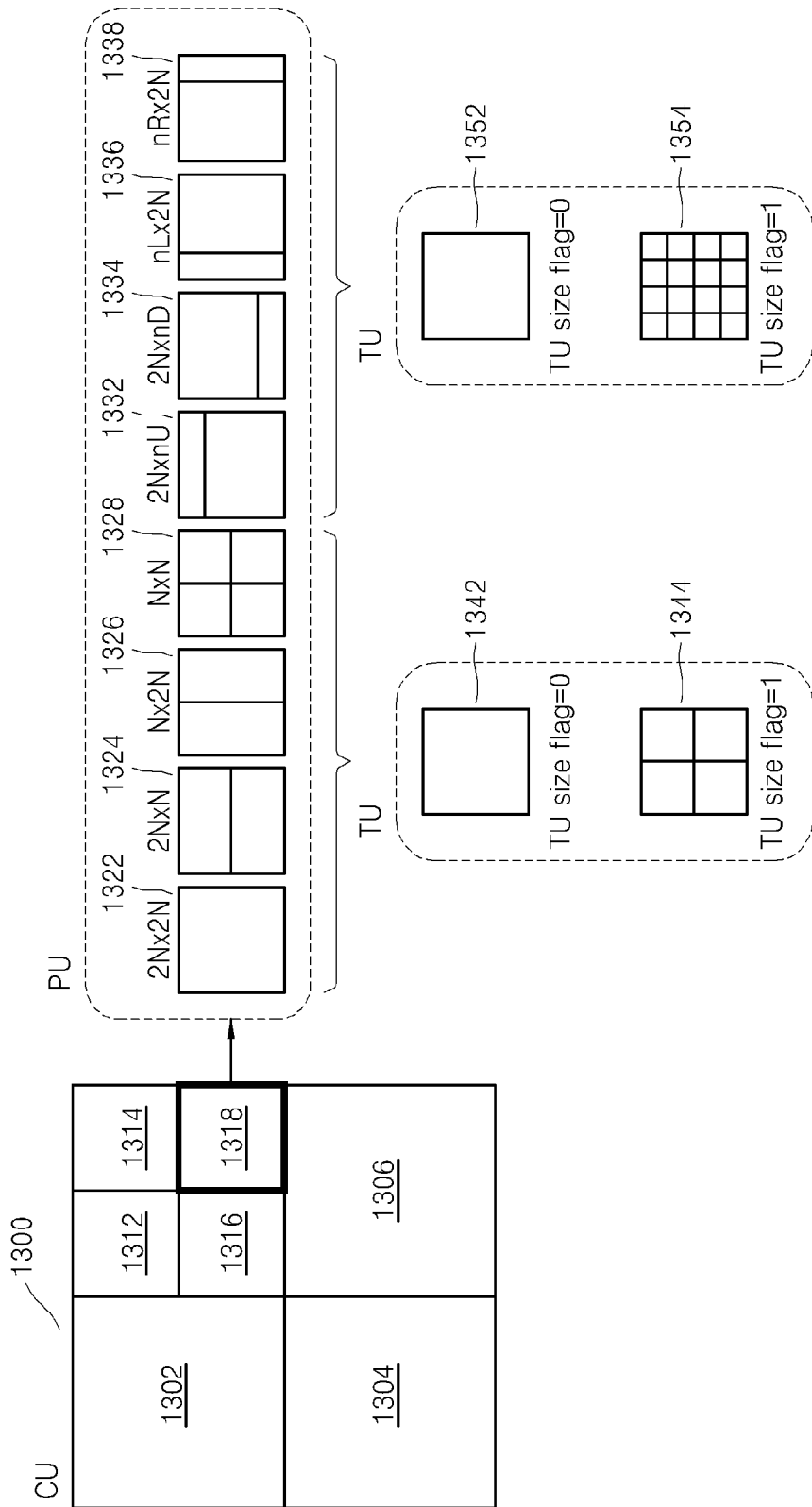
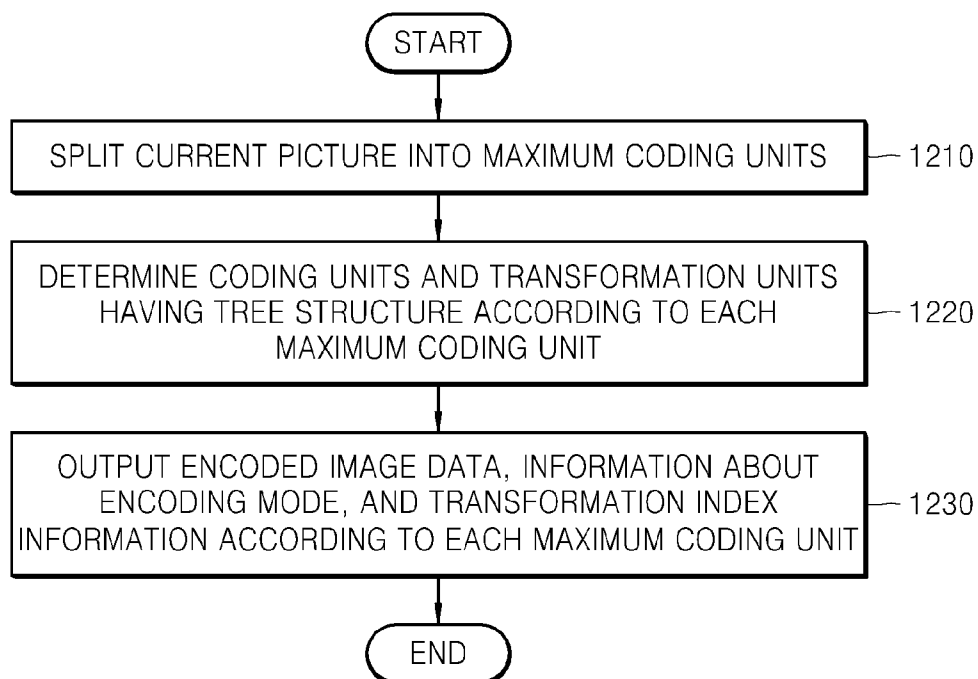
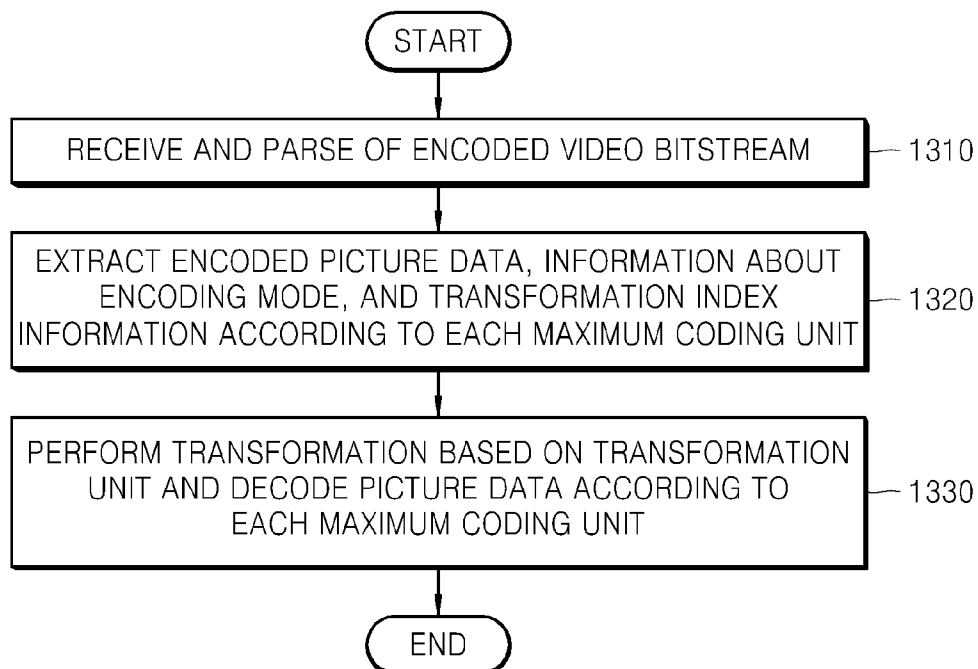




FIG. 21



**FIG. 22****FIG. 23**

**METHOD AND APPARATUS FOR ENCODING  
VIDEO BY USING TRANSFORMATION  
INDEX, AND METHOD AND APPARATUS  
FOR DECODING VIDEO BY USING  
TRANSFORMATION INDEX**

**CROSS-REFERENCE TO RELATED PATENT  
APPLICATION**

[0001] This application is a Continuation application of U.S. patent application Ser. No. 13/079,983, filed on Apr. 5, 2011, in the U.S. Patent and Trademark Office, which claims the benefit of U.S. Provisional Patent Application No. 61/320,826, filed on Apr. 5, 2010, in the U.S. Patent and Trademark Office, and priority from Korean Patent Application No. 10-2010-0096920, filed on Oct. 5, 2010, in the Korean Intellectual Property Office, the disclosures of which are incorporated herein by reference in their entireties.

**BACKGROUND**

[0002] 1. Field

[0003] The exemplary embodiments relate to video encoding and video decoding in which transformation between a spatial domain and a transformation domain is performed.

[0004] 2. Description of the Related Art

[0005] As hardware for reproducing and storing high resolution or high quality video content is being developed and supplied, a need for a video codec for effectively encoding or decoding the high resolution or high quality video content is increasing. In a conventional video codec, a video is encoded according to a limited encoding method based on a macroblock having a predetermined size. In the conventional video codec, video data is encoded and decoded by performing transformation and inverse transformation on macroblocks by using blocks each having the same size.

**SUMMARY**

[0006] According to an aspect of an exemplary embodiment, there is provided a method of encoding a video including: determining a transformation unit that is a data unit in which a current coding unit is transformed, the coding unit being a data unit in which a current picture of the video is encoded; transforming data of the current coding unit based on the determined transformation unit to encode the data of the current coding unit; and outputting the encoded data of the current coding unit, encoding mode information that indicates an encoding mode of the encoded data of the current coding unit, and transformation index information that indicates a structure of the transformation unit transforming the data of the current coding unit, as a bitstream.

[0007] According to another aspect of an exemplary embodiment, there is provided a method of decoding an encoded video including: receiving a bitstream of the encoded video and parsing the received bitstream; extracting encoded data of a current coding unit that is a data unit in which a current picture of the encoded video is encoded, encoding mode information that indicates an encoding mode of the encoded data of the current coding unit, and transformation index information that indicates a structure of a transformation unit that is a data unit in which the data of the current coding unit is transformed, from the parsed bitstream; and performing inverse transformation on the encoded data of the current coding unit based on the transformation index

information to decode the encoded data of the current coding unit transformed in the transformation unit.

[0008] According to another aspect of an exemplary embodiment, there is provided a video encoding apparatus including a processor, the apparatus including: a transformation unit determiner which determines a transformation unit that is a data unit in which a current coding unit is transformed, the coding unit being a data unit in which a current picture of a received video is encoded; an encoding unit which transforms data of the current coding unit based on the determined transformation unit to encode the data of the current coding unit; and an encoded data output unit which outputs the encoded data of the current coding unit, encoding mode information that indicates an encoding mode of the encoded data of the current coding unit, and transformation index information that indicates a structure of the transformation unit transforming the data of the current coding unit.

[0009] According to another aspect of an exemplary embodiment, there is provided a video decoding apparatus including a processor, the apparatus including: a receiver which receives a bitstream of an encoded video and parses the received bitstream; an extractor which extracts encoded data of a current coding unit that is a data unit in which a current picture of the encoded video is encoded, encoding mode information that indicates an encoding mode of the encoded data of the current coding unit, and transformation index information that indicates a structure of a transformation unit that is a data unit in which the current coding unit is transformed, from the parsed bitstream; and a decoder which performs inverse transformation on the encoded data of the current coding unit based on the transformation index information to decode the encoded data of the current coding unit transformed in the transformation unit.

[0010] According to another aspect of an exemplary embodiment, there is provided a video encoding apparatus including a processor, the apparatus including: a maximum coding unit splitter which splits a current picture into at least one maximum coding unit; a coding unit determiner which determines coding units having a tree structure that include coding units of a coded depth that are hierarchical according to depths in a same region of the at least one maximum coding unit and independent in different regions, by independently determining a coding unit of a coded depth to output an encoding result for each deeper coding unit, from among all deeper coding units hierarchically constructed according to depths that indicate numbers of times the at least one maximum coding unit is spatially split, for each of the at least one maximum coding unit, and determining a transformation unit being a data unit in which a current coding unit from among the coding units having the tree structure is transformed, to encode the current coding unit by including transformation based on the transformation unit; and an output unit which encodes and outputs encoded data of the current picture, information about a coded depth of the coding units having the tree structure and an encoding mode, and transformation index information about a structure of transformation units of the coding units of the coded depth, for each of the maximum coding units.

[0011] According to another aspect of an exemplary embodiment, there is provided a video decoding apparatus including a processor, the apparatus including: a receiver which receives a bitstream of an encoded video and parses the received bitstream; an image data and encoded information extractor which extracts encoded data of a picture, informa-

tion about a coded depth and an encoding mode, and transformation index information about a structure of transformation units of the coding units of the coded depth, according to coding units having a tree structure included in each of a plurality of maximum coding units into which the picture is split, from the parsed bitstream; and an image data decoder that decodes the encoded data by performing inverse transformation on the coding units of the coded depth, based on transformation units obtained based on the transformation index information, for each of the plurality of maximum coding units, wherein the coding units having the tree structure comprise coding units corresponding to a coded depth determined to output an encoding result from among deeper coding units hierarchically constructed according to depths that indicate numbers of times the maximum coding unit is spatially split, when at least one of the plurality of maximum coding units is encoded.

[0012] According to another aspect of an exemplary embodiment, there is provided a computer readable recording medium having recorded thereon a program for executing the method of encoding a video.

[0013] According to another aspect of an exemplary embodiment, there is provided a computer readable recording medium having recorded thereon a program for executing the method of decoding a video.

#### BRIEF DESCRIPTION OF THE DRAWINGS

[0014] The above and/or other aspects will become more apparent by describing in detail exemplary embodiments thereof with reference to the attached drawings in which:

[0015] FIG. 1 is a block diagram of a video encoding apparatus using a transformation index, according to an exemplary embodiment;

[0016] FIG. 2 is a block diagram of a video decoding apparatus using a transformation index, according to an exemplary embodiment;

[0017] FIG. 3 is a diagram illustrating the structure of a transformation unit and a transformation index, according to an exemplary embodiment;

[0018] FIG. 4 is a diagram illustrating the structure of a transformation unit and a transformation index, according to an exemplary embodiment;

[0019] FIGS. 5 and 6 are diagrams illustrating examples of use of the transformation index according to an exemplary embodiment;

[0020] FIG. 7 is a flowchart illustrating a video encoding method using a transformation index, according to an exemplary embodiment;

[0021] FIG. 8 is a flowchart illustrating a video decoding method using a transformation index, according to an exemplary embodiment;

[0022] FIG. 9 is a block diagram of a video encoding apparatus using coding units having a tree structure and a transformation index, according to an exemplary embodiment;

[0023] FIG. 10 is a block diagram of a video decoding apparatus using coding units having a tree structure and a transformation index, according to an exemplary embodiment;

[0024] FIG. 11 is a diagram for describing a concept of coding units according to an exemplary embodiment;

[0025] FIG. 12 is a block diagram of an image encoder based on coding units according to an exemplary embodiment;

[0026] FIG. 13 is a block diagram of an image decoder based on coding units according to an exemplary embodiment;

[0027] FIG. 14 is a diagram illustrating deeper coding units according to depths, and partitions according to an exemplary embodiment;

[0028] FIG. 15 is a diagram for describing a relationship between a coding unit and transformation units, according to an exemplary embodiment;

[0029] FIG. 16 is a diagram for describing encoding information of coding units corresponding to a coded depth, according to an exemplary embodiment;

[0030] FIG. 17 is a diagram of deeper coding units according to depths, according to an exemplary embodiment;

[0031] FIGS. 18 through 20 are diagrams for describing a relationship between coding units, prediction units, and transformation units, according to an exemplary embodiment;

[0032] FIG. 21 is a diagram for describing a relationship between a coding unit, a prediction unit or partition, and a transformation unit, according to encoding mode information of Table 1;

[0033] FIG. 22 is a flowchart illustrating a video encoding method that uses a transformation index on the basis of coding units and transformation units having a tree structure, according to an exemplary embodiment; and

[0034] FIG. 23 is a flowchart illustrating a video decoding method that use a transformation index on the basis of the coding units and the transformation units having a tree structure, according to an exemplary embodiment.

#### DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENTS

[0035] Hereinafter, the exemplary embodiments will be described more fully with reference to the accompanying drawings, in which THE exemplary embodiments are shown. In the exemplary embodiments, “unit” may or may not refer to a unit of size, depending on its context, and an ‘image’ may denote a still image for a video or a moving image, that is, the video itself.

[0036] Hereinafter, a ‘coding unit’ is an encoding data unit in which the image data is encoded at an encoder side and an encoded data unit in which the encoded image data is decoded at a decoder side, according to exemplary embodiments. Also, a ‘coded depth’ indicates a depth where a coding unit is encoded.

[0037] Encoding and decoding of a video by using a transformation index on the basis of transformation units having a tree structure, according to an exemplary embodiment, will be described with reference to FIGS. 1 through 8. Encoding and decoding of a video by using a transformation index on the basis of coding units and transformation units having a tree structure, according to an exemplary embodiment, be described with reference to FIGS. 9 through 23.

[0038] First, a method and apparatus for encoding video by using a transformation index and a method and apparatus for decoding video by using a transformation index according to an exemplary embodiment will be described with reference to FIGS. 1 to 8.

[0039] FIG. 1 is a block diagram of a video encoding apparatus 10 using a transformation index, according to an exemplary embodiment.

[0040] Referring to FIG. 1, the video encoding apparatus 10 using a transformation index includes a transformation unit determiner 12, an encoding unit 14, and an encoded data

output unit **16**. For convenience of explanation, the video encoding apparatus using a transformation index will be referred to as a video encoding apparatus **10**. The operations of the transformation unit determiner **12**, the encoding unit **14**, and the encoded data output unit **16** of the video encoding apparatus **10** may be organically controlled by a (not shown) video encoding processor, a central processing unit (CPU), a graphics processing unit (GPU), or the like.

**[0041]** The video encoding apparatus **10** splits a current picture of a received video into data units, each unit having a predetermined size, and performs encoding on each of the data units, to encode the current picture. Hereinafter, a data unit in which the current picture is encoded is referred to as a 'coding unit'. The video encoding apparatus **10** may encode the current picture by performing predictive encoding including inter-prediction and intra-prediction, transformation and quantization, and entropy encoding on each coding unit.

**[0042]** The transformation unit determiner **12** determines a transformation unit, that is, a data unit in which a current coding unit, from among the coding units of the current picture, is transformed. The transformation unit may be included in the current coding unit, and may be determined to be a data unit having a size smaller than or equal to the current coding unit. The transformation unit determiner **12** may generate the transformation unit by halving the height and width of the current coding unit, i.e., quartering the current coding unit.

**[0043]** The transformation unit determiner **12** may generate transformation units of a lower level by halving the height and width of the transformation unit. The transformation unit determiner **12** may split the current coding unit into transformation units each having the same size by splitting all transformation units into transformation units of a lower level. Since the height and width of each transformation unit are halved, the current coding unit may be split into transformation units each having the same size, and the total number of transformation units of a lower level is 4 to the power of a positive integer.

**[0044]** The transformation unit determiner **12** may determine transformation units having a tree structure, to transform the current coding unit. The transformation units having a tree structure include final transformation units whose transformation results are determined to be output, from among the transformation units of the current coding unit.

**[0045]** In order to determine the transformation units having a tree structure, the transformation unit determiner **12** may generate transformation units of a lower level by repeatedly halving the height and width of a transformation unit from among the transformation units of the current coding unit. The transformation unit determiner **12** may determine whether to split each transformation unit into transformation units of lower level, independently from the other transformation units on the same level as that of the each transformation unit.

**[0046]** The transformation unit determiner **12** may select transformation units of a level where a transformation error is minimized, by repeatedly transforming transformation units of each of all levels having a hierarchical structure in the current coding unit. The transformation unit of the level allowing for a minimum transformation error may be determined to be a final transformation unit whose transformation result is to be output. Accordingly, the transformation unit determiner **12** may determine transformation units having a tree structure according to an exemplary embodiment in

which the final transformation units whose transformation results are determined to be output are included.

**[0047]** The encoding unit **14** encodes the current coding unit by generating residual data of the current coding unit and transforming the residual data based on the transformation units determined by the transformation unit determiner **12**. 'Transformation' for video encoding according to an exemplary embodiment denotes a data processing technique for transforming the data of a spatial domain of a video into the data of a transformation domain thereof. Examples of the transformation performed for video encoding by the encoding unit **14** may include frequency transformation, orthogonal transformation, integer transformation, and the like.

**[0048]** The encoded data output unit **16** outputs encoded data of the current coding unit, information that indicates an encoding mode, and transformation index information, as a bitstream.

**[0049]** The information about an encoding mode may include information about various methods and the like used to encode the current coding unit.

**[0050]** The transformation index information may be information about the structure of a transformation unit used to transform the current coding unit. For example, the transformation index information may include the number of times the current coding unit is split to turn into transformation units of a final level, and information about the sizes and shapes of the transformation units.

**[0051]** The transformation index information may represent whether a current transformation unit is split into transformation units of a lower level. For example, a transformation unit split bit corresponding to 1-bit data, that indicates whether a current transformation unit is split into transformation units of a lower level, may be used as the transformation index information.

**[0052]** Transformation index information according to a first exemplary embodiment may indicate whether the current transformation unit is split into transformation units each having the same size. For example, the transformation index information according to the first exemplary embodiment may indicate whether the height and width of the current coding unit are each halved once to obtain four transformation units or are each halved twice to obtain 16 transformation units. In other words, the transformation index information according to the first exemplary embodiment may represent the number of 4 to the power of a positive integer of the transformation units each having the same size, into which the current coding unit is split.

**[0053]** Transformation index information according to a second exemplary embodiment may indicate whether the current coding unit is split into transformation units having various sizes according to a tree structure, according to an exemplary embodiment.

**[0054]** The size of a transformation unit may be determined based on the transformation index and a prediction unit type or a partition type of a current coding unit. For example, the size of a transformation unit corresponding to the transformation index may vary according to a prediction unit type or a partition type of a coding unit.

**[0055]** If the current transformation unit is split into transformation units each having the same size, the current size of the transformation unit may be determined based on the transformation index and a prediction unit type or a partition type of a corresponding coding unit. Also, if a current coding unit is split into transformation units having various sizes accord-

ing to a tree structure, the size of a current transformation unit may be determined based on the transformation index and a prediction unit type or a partition type of the current coding unit.

[0056] According to an exemplary embodiment, the maximum size of a transformation unit may be equal to the size of a current coding unit. According to other exemplary embodiment, the maximum size of a transformation unit may be determined based on a prediction unit type or a partition type of a current prediction unit or partition. For example, the maximum size of the current transformation unit size may indicate the size of a maximum square included in the current prediction unit or partition.

[0057] For example, the transformation index information according to the second exemplary embodiment may be represented as a bitstring obtained by arranging transformation unit split bits of transformation units of respective levels that are obtained until the current coding unit is split into transformation units having the tree structure. The transformation index information according to the second exemplary embodiment may include a bitstring obtained by arranging transformation unit split bits of adjacent transformation units of the same level in a sequence where the transformation units are scanned in a zigzag manner. When a predetermined transformation unit may be split into transformation units of a lower level having a hierarchical structure, the transformation index information according to the second exemplary embodiment may include a bitstring obtained by arranging transformation unit split bits of the transformation units of the lower level in a sequence where the transformation units are scanned in a zigzag manner.

[0058] FIG. 2 is a block diagram of a video decoding apparatus using a transformation index, according to an exemplary embodiment

[0059] Referring to FIG. 2, the video decoding apparatus 20 using a transformation index includes a receiver 22, an extractor 24, and a decoder 26. For convenience of explanation, the video decoding apparatus 20 using a transformation index will be referred to as a video decoding apparatus 20. The operations of the receiver 22, the extractor 24, and the decoder 26 of the video decoding apparatus 20 may be controlled by a (not shown) video decoding processor, a CPU, a GPU, or the like.

[0060] To decode a current picture of a received video, the video decoding apparatus 20 may decode the current picture by performing entropy decoding, dequantization, inverse transformation, and predictive decoding, including inter-prediction/compensation and intra-prediction/compensation, on each coding unit of the current picture.

[0061] The receiver 22 receives and parses a bitstream for an encoded video. The extractor 24 extracts encoded data of each coding unit of the current picture, information about an encoding mode, and transformation index information of a current coding unit, from the bitstream parsed by the receiver 22.

[0062] The decoder 26 decodes the encoded data by generating transformation units of the current coding unit according to the transformation index information extracted by the extractor 24 and performing inverse transformation on the current coding unit on the basis of the transformation units. As a result of the decoding of coding units, the current picture may be restored.

[0063] The concept of the transformation unit is similar to that described above with reference to the video encoding

apparatus 10 of FIG. 1. In other words, a transformation unit according to an exemplary embodiment may be a transformation unit obtained by halving the height and width of the current coding unit or a transformation unit of an upper level. All of the transformation units included in the current coding unit according to an exemplary embodiment may have the same size. A transformation unit according to another exemplary embodiment is a transformation unit of one level from among the transformation units of the current coding unit that have a tree structure, and may be split into transformation units of a lower level through repetitive splitting for each level of transformation unit or into transformation units of a lower level through independent splitting between adjacent transformation units.

[0064] The decoder 26 may read information that indicates the number of times the current coding unit is split to turn into transformation units of a final level, information about the sizes and shapes of the transformation units, and the like, from the transformation index information.

[0065] The decoder 26 may read information that indicates whether a current transformation unit is split into transformation units of a lower level, from the transformation index information.

[0066] The decoder 26 may read information that indicates the level of a transformation unit from a transformation index according to a first exemplary embodiment. In this case, since the current coding unit is split into transformation units of the same sizes for each level until transformation units of a final level are obtained, the decoder 26 may determine the transformation units of the final level having the same size by determining the final level of transformation units according to the transformation index and halving the heights and widths of all of the transformation units of an upper level when the current coding unit is split into the transformation units of the final level.

[0067] The decoder 26 may determine the size of a transformation unit based on the transformation index and a prediction unit type or a partition type of a current coding unit. For example, the size of a transformation unit corresponding to the transformation index may vary according to a prediction unit type or a partition type of a coding unit.

[0068] If the decoder 26 may split a current coding unit and then determine the current transformation units having the same size, the size of the current transformation unit may be determined based on the transformation index and a prediction unit type or a partition type of the current coding unit. Also, if a current coding unit is split into transformation units having various sizes according to a tree structure, a size of a current transformation unit may be determined based on the transformation index and a prediction unit type or a partition type of the current coding unit.

[0069] The decoder 26 may determine transformation units obtained according to a tree structure, based on transformation index information according to a second exemplary embodiment. For example, a bitstring of the transformation index information according to the second exemplary embodiment may be a bitstring obtained by arranging transformation unit split bits for transformation units of each level. The decoder 26 may determine transformation units into which the current coding unit is split according to a tree structure, by reading the bitstring of the transformation index information according to the second exemplary embodiment and splitting the current coding unit so that independent split-

ting is performed between transformation units on the same level and that repetitive splitting is performed between levels.

[0070] At this time, the decoder 26 may read transformation unit split bits of the transformation units on the same level from the transformation index information according to the second exemplary embodiment and may split transformation units on an upper level into transformation units on a lower level in a zigzag scan sequence. When a transformation unit of an upper level is split into transformation units of a lower level, the decoder 26 may read the transformation unit split bits of the transformation units of the lower level, which are included in the transformation unit of the upper level, in a zigzag scan sequence.

[0071] The inverse transformation performed for video decoding by the decoder 26 may be referred to as a process of transforming data of a transformation domain into data of a spatial domain. Examples of the inverse transformation performed by the decoder 26 may include frequency inverse-transformation, orthogonal inverse-transformation, integer inverse-transformation, and the like.

[0072] The video encoding apparatus 10 and the video decoding apparatus 20 may efficiently encode and decode information used to determine various sizes and shapes of transformation units having a tree structure, which are necessary for performing transformation and inverse transformation during video encoding and decoding based on various sizes and shapes of coding units, by using the transformation index according to an exemplary embodiment.

[0073] FIG. 3 is a diagram illustrating the structure of a transformation unit and a transformation index, according to a first exemplary embodiment.

[0074] A transformation unit group 32 of level 0, a transformation unit group 34 of level 1, and a transformation unit group 36 of level 2 are illustrated according to a transformation index, as the transformation unit structure according to the first embodiment that includes transformation units of the same sizes in order to transform a coding unit CU0 30. The transformation index according to the first exemplary embodiment may represent the number of times the coding unit CU0 30 is split to turn into a transformation unit group of a current level, that is, a level number.

[0075] In other words, the transformation unit group 32 of level 0 includes a transformation unit TU0 that has the same size as the coding unit CU0 30 by splitting the height and width of the coding unit CU0 30 zero times. In this case, the transformation index of the transformation unit group 32 of level 0 is 0.

[0076] The transformation unit group 34 of level 1 includes transformation units TU10, TU11, TU12, and TU13 each having a height and a width being half of those of the coding unit CU0 30, by splitting the height and width of the coding unit CU0 30 once. In this case, the transformation index of the transformation unit group 34 of level 1 is 1.

[0077] The transformation unit group 36 of level 2 includes transformation units TU20, TU21, TU22, TU23, TU24, TU25, TU26, TU27, TU28, TU29, TU2A, TU2B, TU2C, TU2D, TU2E, and TU2F each having a height and size being a quarter of those of the coding unit CU0 30, by splitting the height and width of the coding unit CU0 30 twice. In this case, the transformation index of the transformation unit group 36 of level 2 is 2.

[0078] FIG. 4 is a diagram illustrating the structure of a transformation unit and a transformation index, according to a second exemplary embodiment.

[0079] A transformation unit group 40 is illustrated as the transformation unit structure according to the second exemplary embodiment that includes transformation units having a tree structure in order to transform the coding unit CU0 30. The transformation index according to the second exemplary embodiment may be represented as a bitstring of transformation unit split bits for each level that is used until the transformation units having a tree structure are obtained from the coding unit CU0 30.

[0080] In other words, since a transformation unit of level 0 having the same size as the coding unit CU0 30 is divided into transformation units TU40, TU41, TU42, and TU43 of level 1, a transformation unit split bit 1 for level 1 may be generated and included in the transformation index. Since transformation unit split bits for transformation units on the same level are arranged in a zigzag scanning sequence, transformation unit split bits for level 0 may be arranged in a sequence of transformation unit split bits for the transformation units TU40, TU41, TU42, and TU43. Since the transformation units TU40 and TU41 are not split, transformation unit split bits 0 and 0 for the transformation units TU40 and TU41 may be sequentially generated and included in the transformation index.

[0081] The transformation unit TU42 of level 1 is further split into transformation units TU50, TU51, TU52, and TU53 of level 2. Accordingly, a transformation unit split bit 1 for the transformation unit TU42 of level 1 may be generated. When a transformation unit of a current level is split into transformation units of a lower level, transformation unit split bits for the transformation units of the lower level may be included in the transformation index. Accordingly, a transformation unit split bit 0 for the transformation unit TU50 of level 2, a transformation unit split bit 1 for the transformation unit TU51 of level 2, a transformation unit split bit 0 for the transformation unit TU52 of level 2, and a transformation unit split bit 0 for the transformation unit TU53 of level 2 may be sequentially included in the transformation index. The transformation unit TU51 of level 2 is further split into transformation units TU60, TU61, TU62, and TU63 of level 3. A transformation unit of level 3 is a minimum transformation unit or a minimum unit, and the transformation unit of level 3 is not further split.

[0082] In other words, if a transformation unit split bit for a transformation unit of a current level is 1 and transformation units of a lower level are not the minimum transformation unit or the minimum unit, transformation unit split bits for the transformation units of the lower level may be consecutively arranged after the transformation unit split bit for the transformation unit of the current level.

[0083] Lastly, since the transformation unit TU43 of level 1 is not split, the transformation unit split bit 0 may be included in the transformation index.

[0084] Accordingly, the transformation index according to the second exemplary embodiment for the coding unit CU0 30 may be determined to be 1, 0, 0, 1, 0, 1, 0, 0, 0. If transformation unit split bits for transformation units of a lower level are consecutively 0, it may be understood that the transformation units of the lower level are not further split.

[0085] The transformation unit split bits generated in this way are arranged, starting from the transformation unit of level 0 having the same size as the coding unit, in such a way that transformation unit split bits of transformation units on the same level are arranged in a zigzag scanning sequence and that if a transformation unit of a predetermined level is split

into transformation units of a lower level having a hierarchical structure, transformation unit split bits for the transformation units of the lower level are arranged in a zigzag scanning sequence, whereby the transformation index according to the second exemplary embodiment may be determined.

[0086] FIGS. 5 and 6 are diagrams illustrating examples of use of the transformation index according to the second exemplary embodiment.

[0087] As described above with reference to FIG. 4, if a transformation unit is not further split, transformation unit split bits are not further generated. Thus, the transformation index according to the second exemplary embodiment may be set according to the size of the minimum transformation unit or the minimum unit.

[0088] When a minimum transformation unit 52 of a coding unit 50 of a  $2N \times 2N$  size has a size of  $N \times N$ , the coding unit 50 may be only split until  $N \times N$  transformation units are obtained, and thus a transformation index of the coding unit 50 for a transformation unit group 54 may be set to be 1.

[0089] On the other hand, when a minimum transformation unit 62 of a coding unit 60 of a  $2N \times 2N$  size has a size of  $(N/2) \times (N/2)$ , transformation units of a  $N \times N$  size in a transformation unit group 64 may be each split one more time. Accordingly, a transformation index of the coding unit 60 for the transformation unit group 64 may include, not only a transformation unit split bit 1 for a transformation unit of level 0, but also transformation unit split bits 0, 0, 0, and 0 for the transformation units of level 1 in the transformation unit group 64.

[0090] The transformation units described above with reference to FIGS. 3 through 6 may be determined by the transformation unit determiner 12 of the video encoding apparatus 10, and the transformation index may be encoded by the encoded data output unit 14. The extractor 24 of the video decoding apparatus 20 may extract transformation index information, and the decoder 26 thereof may form a transformation unit by reading the transformation index, and perform inverse transformation on the transformation unit.

[0091] FIG. 7 is a flowchart illustrating a video encoding method using a transformation index, according to an exemplary embodiment.

[0092] In operation 72, a transformation unit, that is, a data unit in which a current coding unit from among coding units of a current picture is transformed, is determined. The transformation unit may be determined to be a data unit having a size smaller than or equal to the current coding unit so that the transformation unit is included in the current coding unit, and the transformation unit may be generated by halving the height and width of the coding unit. Since the transformation unit may form a hierarchical structure, transformation units of a lower level may be generated by halving the height and width of a transformation unit of an upper level. For example, all transformation units of a predetermined level within a current coding unit may be split into transformation units of a lower level, so that as many transformation units of the same size as the number of 4 to the power of a positive integer may be generated.

[0093] The transformation unit may include final transformation units whose transformation results are determined to be output, from among the transformation units of the current coding unit.

[0094] The hierarchical structure of transformation units according to an exemplary embodiment may be a tree structure. While the height and width of a current transformation

unit from among the transformation units of the current coding unit are repeatedly halved, it is determined whether the halving is performed independently from the other transformation units, and thus transformation units of a lower level are generated. The transformation units on each level within the same region may form a hierarchical structure. Final transformation units are determined from among the transformation units generated in this way so that transformation results are output, so that transformation units having a tree structure according to an exemplary embodiment may be obtained.

[0095] Transformation units of a level where an error, due to transformation for each transformation unit, is minimized may be selected as the final transformation units whose transformation results are output, by repeatedly performing transformation on each level for transformation units having a hierarchical structure in the current coding unit and comparing the results of the transformations with each other.

[0096] In operation 74, the current coding unit is encoded, including transformation based on the transformation unit. In operation 76, encoded data of the current coding unit, information about an encoding mode, and transformation index information are output.

[0097] The transformation index information about the structure of a transformation unit according to an exemplary embodiment may indicate whether a current transformation unit is split into transformation units of a lower level. The transformation index information about the structure of a transformation unit according to an exemplary embodiment may include the number of times the current coding unit is split to turn into transformation units of a final level, and information about the sizes and shapes of the transformation units.

[0098] Transformation index information according to a first exemplary embodiment may indicate a level that identifies the total number of splitting times from a current coding unit to a coding unit of a final level. The transformation units of each level may have the same sizes.

[0099] Transformation index information according to a second exemplary embodiment may indicate whether the current coding unit is repeatedly split to obtain transformation units having a tree structure. The transformation index information according to the second exemplary embodiment may be in the shape of a bitstring obtained by arranging transformation unit split bits that indicate whether a transformation unit of each level is split into transformation units of a lower level. The bitstring of the transformation index information may be obtained by arranging transformation unit split bits of adjacent transformation units of the same level in a sequence where the transformation units are scanned in a zigzag manner. When a current transformation unit includes transformation units of a lower level having a hierarchical structure, a bitstring of the transformation index information may be determined so that transformation unit split bits of the transformation units of a lower level are arranged in a sequence where the transformation units are scanned in a zigzag manner.

[0100] FIG. 8 is a flowchart illustrating a video decoding method using a transformation index, according to an exemplary embodiment.

[0101] In operation 82, a bitstream for an encoded video is received and parsed.



**[0102]** In operation **84**, encoded data of a current coding unit of a current picture, information about an encoding mode, and transformation index information are extracted from the parsed bitstream.

**[0103]** In operation **86**, inverse transformation is performed on the current coding unit, based on transformation units obtained according to the transformation index information, and thus the encoded data is decoded. As a result of the decoding of each coding unit, the current picture may be restored. Whether a current transformation unit is split into transformation units of a lower level may be read based on the transformation index information about the structure of a transformation unit according to an exemplary embodiment, and the transformation units may be determined. The inverse transformation on the current coding unit may be performed based on the transformation units.

**[0104]** The number of times the current coding unit is split until transformation units of a final level is obtained, and information about the sizes, shapes, and the like, of the transformation units may be read from the transformation index information about the structure of a transformation unit according to an exemplary embodiment.

**[0105]** A level that indicates the total number of splitting times from the current coding unit to a transformation unit of a final level may be read from the transformation index information according to the first exemplary embodiment. The level of a transformation unit indicates the number of times the current coding unit is split to obtain 4 transformation units each having the same size. Accordingly, a structure of transformation units in which the current coding unit is split into transformation units each having the same size may be determined.

**[0106]** A bitstring of transformation unit split bits that indicates whether a transformation unit of each level is split into transformation units of a lower level is read from the transformation index information according to the second exemplary embodiment, until the current coding unit is repeatedly split to obtain transformation units having the tree structure. Thus, a structure of transformation units into which the current coding unit is split according to the tree structure may be determined.

**[0107]** For example, transformation unit split bits of adjacent transformation units on the same level, from among the transformation index information according to the second exemplary embodiment, may be read in a sequence where the transformation units are scanned in a zigzag manner. Moreover, transformation unit split bits of transformation units of a lower level having a hierarchical structure included in a transformation unit of a predetermined level, from among transformation index information according to another exemplary embodiment, may be read in a sequence where the transformation units of the lower level are scanned in a zigzag manner.

**[0108]** First, a method and apparatus for encoding video by using a coding unit having a tree structure and a transformation index, and a method and apparatus for decoding video by using a coding unit having a tree structure and a transformation index, according to an exemplary embodiment will be described with reference to FIGS. **9** to **23**.

**[0109]** FIG. **9** is a block diagram of a video encoding apparatus **100**, which uses a transformation index on the basis of coding units and transformation units having a tree structure, according to an exemplary embodiment.

**[0110]** The video encoding apparatus **100**, which uses the transformation index based on coding units and transmission units having a tree structure, includes a maximum coding unit splitter **110**, a coding unit determiner **120**, and an output unit **130**. For convenience of explanation, the video encoding apparatus **100**, which uses the transformation index based on coding units and transmission units having a tree structure, is referred to as a video encoding apparatus **100**.

**[0111]** The maximum coding unit splitter **110** may split a current picture based on a maximum coding unit for the current picture of an image. If the current picture is larger than the maximum coding unit, image data of the current picture may be split into the at least one maximum coding unit. The maximum coding unit according to an exemplary embodiment may be a data unit having a size of 32×32, 64×64, 128×128, 256×256, etc., wherein a shape of the data unit is a square having a width and height in squares of 2. The image data may be output to the coding unit determiner **120** according to the at least one maximum coding unit.

**[0112]** A coding unit according to an exemplary embodiment may be characterized by a maximum size and a depth. The depth denotes a number of times the coding unit is spatially split from the maximum coding unit, and as the depth deepens or increases, 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 depth may include a plurality of coding units corresponding to lower depths.

**[0113]** As described above, the 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 exemplary embodiment 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.

**[0114]** 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 split may be predetermined.

**[0115]** The coding unit determiner **120** 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 a finally encoded image data according to the at least one split region. In other words, the coding unit determiner **120** determines a coded depth by encoding 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. Thus, the encoded image data of the coding unit corresponding to the determined coded depth is finally output. Also, the coding units corresponding to the coded depth may be regarded as encoded coding units.

**[0116]** The determined coded depth and the encoded image data according to the determined coded depth are output to the output unit **130**.

**[0117]** The image data in the maximum coding unit is encoded based on the deeper coding units corresponding to at least one depth equal to or below 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.

**[0118]** The size of the maximum coding unit is split as a coding unit is hierarchically split according to depths, and as the number of coding units increases. Also, even if coding units correspond to 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 the each coding unit, separately. Accordingly, even when image data is included in one maximum coding unit, the image data is split to 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.

**[0119]** Accordingly, the coding unit determiner **120** may determine coding units having a tree structure included in the maximum coding unit. The 'coding units having a tree structure' according to an exemplary embodiment 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 of 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.

**[0120]** A maximum depth according to an exemplary embodiment is an index related to the number of splitting times from a maximum coding unit to a minimum coding unit. A first maximum depth according to an exemplary embodiment may denote the total number of splits from the maximum coding unit to the minimum coding unit. A second maximum depth according to an exemplary embodiment 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, 5 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.

**[0121]** 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. Transformation may be performed according to method of orthogonal transformation or integer transformation.

**[0122]** 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 descrip-

tion, the prediction encoding and the transformation will now be described based on a coding unit of a current depth, in a maximum coding unit.

**[0123]** The video encoding apparatus **100** may variably 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 different data units may be used for each operation.

**[0124]** 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 to perform the prediction encoding on the image data in the coding unit.

**[0125]** 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 coded depth, i.e., based on a coding unit that is no longer split to 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 be referred to as a 'prediction unit'. A partition obtained by splitting the prediction unit may include a prediction unit or a data unit obtained by splitting at least one of a height and a width of the prediction unit.

**[0126]** 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  $2N \times 2N$ , and 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 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.

**[0127]** 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 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 a least encoding error.

**[0128]** 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.

**[0129]** As described above with reference to FIGS. 1 through 8, 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.

**[0130]** A data unit used as a base of the transformation will be referred to as a 'transformation unit'. A transformation depth that indicates the number of splits 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 also  $2N \times 2N$ , may be 1 when each of the height and width of the current coding unit is split into two equal parts, totally split into  $4^1$  transformation units, and the size of the transformation unit is thus  $N \times N$ , and may be 2 when each of the height and

width of the current coding unit is split into four equal parts, totally split into  $4^2$  transformation units and the size of the transformation unit is thus  $N/2 \times N/2$ . For example, the transformation unit may be set according to a hierarchical tree structure, in which a transformation unit of an upper transformation depth is split into four transformation units of a lower transformation depth according to the hierarchical characteristics of a transformation depth.

**[0131]** Similar 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 having the tree structure according to transformation depths.

**[0132]** 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 coding unit determiner **120** not only determines a coded depth having a 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.

**[0133]** Coding units according to a tree structure in a maximum coding unit and a method of determining a partition, according to exemplary embodiments, will be described in detail later with reference to FIGS. **11** and **12**.

**[0134]** The coding unit determiner **120** may measure an encoding error of deeper coding units according to depths by using Rate-Distortion Optimization based on Lagrangian multipliers.

**[0135]** The output unit **130** outputs the image data of the maximum coding unit, which is encoded based on the at least one coded depth determined by the coding unit determiner **120**, and information about the encoding mode according to the coded depth, in bitstreams.

**[0136]** The encoded image data may be obtained by encoding residual data of an image.

**[0137]** The information about the encoding mode according to coded depth may include information about the coded depth, about the partition type in the prediction unit, the prediction mode, and the size of the transformation unit.

**[0138]** 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.

**[0139]** 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.

**[0140]** 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.

**[0141]** Accordingly, the output unit **130** 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.

**[0142]** The minimum unit according to an exemplary embodiment is a rectangular data unit obtained by splitting the minimum coding unit constituting the lowermost depth by 4. Alternatively, the minimum unit may be a maximum rectangular 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.

**[0143]** For example, the encoding information output through the output unit **130** 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 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 be inserted into SPS (Sequence Parameter Set) or a header of a bitstream. Furthermore, the encoding information output through the output unit **130** may include transformation index information about a structure of a transformation unit according to an exemplary embodiment, as described above with reference to FIGS. **1** through **8**.

**[0144]** 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, the coding unit of the current depth having the size of  $2N \times 2N$  may include a maximum of 4 coding units of the lower depth.

**[0145]** 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 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.

**[0146]** Thus, if an image having high resolution or 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 com-

pression 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.

[0147] FIG. **10** is a block diagram of a video decoding apparatus **200** using coding units having a tree structure and a transformation index, according to an exemplary embodiment.

[0148] The video decoding apparatus **200** includes a receiver **210**, an image data and encoding information extractor **220**, and an image data decoder **230**. 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. **9** and the video encoding apparatus **100**.

[0149] The receiver **210** receives and parses a bitstream of an encoded video. The image data and encoding information extractor **220** extracts encoded image data for each coding unit from the parsed bitstream, wherein the coding units have a tree structure according to each maximum coding unit, and outputs the extracted image data to the image data decoder **230**. The image data and encoding information extractor **220** may extract information about a maximum size of a coding unit of a current picture, from a header about the current picture or SPS.

[0150] Also, the image data and encoding information extractor **220** extracts information about a coded depth and an encoding mode 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 image data decoder **230**. In other words, the image data in a bit stream is split into the maximum coding unit so that the image data decoder **230** decodes the image data for each maximum coding unit.

[0151] 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. Furthermore, the image data and encoding information extractor **220** may extract transformation index information about a structure of a transformation unit according to an embodiment as described above with reference to FIGS. **1** through **8** as the extracted information about the coded depth and the encoding mode.

[0152] The information about the coded depth and the encoding mode according to each maximum coding unit extracted by the image data and encoding information extractor **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.

[0153] 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 image data and encoding information extractor **220** may extract the information about the coded depth and the encoding mode according to the 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.

[0154] The image data 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 image data 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 a prediction including intra prediction and motion compensation, and an inverse transformation. Inverse transformation may be performed according to method of inverse orthogonal transformation or inverse integer transformation.

[0155] The image data 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.

[0156] Also, the image data 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, to perform the inverse transformation according to maximum coding units.

[0157] The image data 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 image data decoder **230** may decode encoded data of at least one coding unit corresponding to the each coded depth in 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 for each coding unit corresponding to the coded depth, and output the image data of the current maximum coding unit.

[0158] In other words, data units containing the encoding information that indicates 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 image data decoder **230** in the same encoding mode.

[0159] 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, the coding units having the tree structure determined to be the optimum coding units in each maximum coding unit may be decoded. Also, the maximum size of coding unit is determined considering resolution and an amount of image data.

[0160] Accordingly, even if image data has 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.

[0161] A method of determining coding units having a tree structure, a prediction unit, and a transformation unit, according to an exemplary embodiment, will now be described with reference to FIGS. 11 through 21.

[0162] FIG. 11 is a diagram for describing a concept of hierarchical coding units according to an exemplary embodiment.

[0163] A size of a coding unit may be expressed in width×height, and may be 64×64, 32×32, 16×16, and 8×8. A coding unit of 64×64 may be split into partitions of 64×64, 64×32, 32×64, or 32×32, and a coding unit of 32×32 may be split into partitions of 32×32, 32×16, 16×32, or 16×16, a coding unit of 16×16 may be split into partitions of 16×16, 16×8, 8×16, or 8×8, and a coding unit of 8×8 may be split into partitions of 8×8, 8×4, 4×8, or 4×4.

[0164] In video data 310, a resolution is 1920×1080, a maximum size of a coding unit is 64, and a maximum depth is 2. In video data 320, a resolution is 1920×1080, a maximum size of a coding unit is 64, and a maximum depth is 3. In video data 330, a resolution is 352×288, a maximum size of a coding unit is 16, and a maximum depth is 1. The maximum depth shown in FIG. 11 denotes a total number of splits from a maximum coding unit to a minimum decoding unit.

[0165] If a resolution is high or a data amount is large, a maximum size of a coding unit may be large to not only increase encoding efficiency but also to 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.

[0166] 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 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 coding unit once.

[0167] 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.

[0168] FIG. 12 is a block diagram of an image encoder 400 based on coding units, according to an exemplary embodiment.

[0169] The image encoder 400 performs operations of the coding unit determiner 120 of the video encoding apparatus 100 to encode image data. In other words, an intra predictor 410 performs intra prediction on coding units in an intra mode, from among a current frame 405, and a motion estimator 420 and a motion compensator 425 performs inter estimation and motion compensation on coding units in an

inter mode from among the current frame 405 by using the current frame 405, and a reference frame 495.

[0170] 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.

[0171] In order for the image encoder 400 to be applied in the video encoding apparatus 100, all elements of the image 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 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.

[0172] Specifically, the intra predictor 410, the motion estimator 420, and the motion compensator 425 determines 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.

[0173] FIG. 13 is a block diagram of an image decoder 500 based on coding units, according to an exemplary embodiment.

[0174] 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 is output as inverse quantized data through an entropy decoder 520 and an inverse quantizer 530, and the inverse quantized data is restored to image data in a spatial domain through an inverse transformer 540.

[0175] 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.

[0176] The image data in the spatial domain, which 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 that is post-processed through the deblocking unit 570 and the loop filtering unit 580 may be output as the reference frame 585.

[0177] In order to decode the image data in the image data decoder 230 of the video decoding apparatus 200, the image decoder 500 may perform operations that are performed after the parser 510.

[0178] In order for the image decoder 500 to be applied in the video decoding apparatus 200, all elements of the image 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.

[0179] Specifically, the intra predictor **550** and the motion compensator **560** perform operations based on partitions and a prediction mode for each of the coding units having a tree structure, and the inverse transformer **540** perform operations based on a size of a transformation unit for each coding unit.

[0180] FIG. 14 is a diagram illustrating deeper coding units according to depths, and partitions, according to an exemplary embodiment.

[0181] The video encoding apparatus **100** and the video decoding apparatus **200** use hierarchical coding units 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 individually set according to an input of a user. Sizes of deeper coding units according to depths may be determined according to the predetermined maximum size of the coding unit.

[0182] In a hierarchical structure **600** of coding units, according to an exemplary embodiment, 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**.

[0183] 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 64×64. The depth deepens along the vertical axis, and a coding unit **620** having a size of 32×32 and a depth of 1, a coding unit **630** having a size of 16×16 and a depth of 2, a coding unit **640** having a size of 8×8 and a depth of 3, and a coding unit **650** having a size of 4×4 and a depth of 4 exist. The coding unit **650** having the size of 4×4 and the depth of 4 is a minimum coding unit.

[0184] 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 64×64 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 64×64, partitions **612** having the size of 64×32, partitions **614** having the size of 32×64, or partitions **616** having the size of 32×32.

[0185] Similarly, a prediction unit of the coding unit **620** having the size of 32×32 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 32×32, partitions **622** having a size of 32×16, partitions **624** having a size of 16×32, and partitions **626** having a size of 16×16.

[0186] Similarly, a prediction unit of the coding unit **630** having the size of 16×16 and the depth of 2 may be split into partitions included in the coding unit **630**, i.e., a partition having a size of 16×16 included in the coding unit **630**, partitions **632** having a size of 16×8, partitions **634** having a size of 8×16, and partitions **636** having a size of 8×8.

[0187] Similarly, a prediction unit of the coding unit **640** having the size of 8×8 and the depth of 3 may be split into partitions included in the coding unit **640**, i.e. a partition having a size of 8×8 included in the coding unit **640**, partitions **642** having a size of 8×4, partitions **644** having a size of 4×8, and partitions **646** having a size of 4×4.

[0188] The coding unit **650** having the size of 4×4 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 4×4. Also, a prediction unit of the coding unit **650** may be split into partitions **652** having a size of 4×2, partitions **654** having a size of 2×4, and partitions **656** having a size of 2×2.

[0189] In order to determine the at least one coded depth of the coding units constituting the maximum coding unit **610**, the coding unit determiner **120** of the video encoding apparatus **100** performs encoding for coding units corresponding to each depth included in the maximum coding unit **610**.

[0190] A 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.

[0191] 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, by 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 coding unit **610** may be selected as the coded depth and a partition type of the coding unit **610**.

[0192] FIG. 15 is a diagram for describing a relationship between a coding unit **710** and transformation units **720**, according to an exemplary embodiment.

[0193] The video encoding apparatus **100** or **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 transformation units for transformation during encoding may be selected based on data units that are not larger than a corresponding coding unit.

[0194] For example, in the video encoding apparatus **100** or **200**, if a size of the coding unit **710** is 64×64, transformation may be performed by using the transformation units **720** having a size of 32×32.

[0195] Also, data of the coding unit **710** having the size of 64×64 may be encoded by performing the transformation on each of the transformation units having the size of 32×32, 16×16, 8×8, and 4×4, which are smaller than 64×64, and then a transformation unit having the least coding error may be selected.

[0196] FIG. 16 is a diagram for describing encoding information of coding units corresponding to a coded depth, according to an exemplary embodiment.

[0197] The 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 corresponding to a coded depth, as information about an encoding mode.

[0198] 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  $2N \times 2N$  may be split into any one of a partition **802** having a size of  $2N \times 2N$ , a partition **804** having a size of  $2N \times N$ , a partition **806** having a size of  $N \times 2N$ , and a partition **808** having a size of  $N \times N$ . Here, the information **800** about a partition type is set to indicate one of the partition **804** having a size of  $2N \times N$ , the partition **806** having a size of  $N \times 2N$ , and the partition **808** having a size of  $N \times N$ .

[0199] 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**.

[0200] 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 intra transformation unit **828**. Also, the encoding information may include transformation index information about a structure of a transformation unit according.

[0201] The image data and encoding information extractor **220** of the video decoding apparatus **200** may extract and use the information **800**, **810**, and **820** for decoding, according to each deeper coding unit.

[0202] FIG. 17 is a diagram of deeper coding units according to depths, according to an exemplary embodiment.

[0203] 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.

[0204] A prediction unit **910** for prediction encoding 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 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.

[0205] 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  $2N_0 \times 2N_0$ .

[0206] Errors of encoding including the prediction encoding in the partition types **912** through **918** are compared, and the least encoding error is determined among the partition types. If an encoding error is smallest in one of the partition types **912** through **916**, the prediction unit **910** may not be split into a lower depth.

[0207] If the encoding error is the smallest in the partition type **918**, a depth is changed from 0 to 1 to split the partition type **918** in operation **920**, and encoding is repeatedly per-

formed on coding units **930** having a depth of 2 and a size of  $N_0 \times N_0$  to search for a minimum encoding error.

[0208] A prediction unit **940** for prediction encoding the 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  $N_1 \times N_1$ .

[0209] If an encoding error is the smallest in the partition type **948**, 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.

[0210] When a maximum depth is d, 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  $N_{(d-1)} \times N_{(d-1)}$ .

[0211] 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.

[0212] Even when the partition type **998** 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 a 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 and a minimum coding unit **980** having a lowermost depth of d-1 is no longer split to a lower depth, split information for the minimum coding unit **980** is not set.

[0213] A data unit **999** may be a 'minimum unit' for the current maximum coding unit. A minimum unit according to an exemplary embodiment may be a rectangular data unit obtained by splitting a 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 partition type and a prediction mode as an encoding mode of the coded depth.

[0214] 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 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.

[0215] The image data and encoding information extractor 220 of the video decoding apparatus 200 may extract and use the information about the coded depth and the prediction unit of the coding unit 900 to decode the partition 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.

[0216] FIGS. 18, 19, and 20 are diagrams for describing a relationship between coding units 1010, prediction units 1060, and transformation units 1070, according to an exemplary embodiment.

tion, motion estimation, motion compensation, transformation, and inverse transformation individually on a data unit in the same coding unit.

[0221] 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 and decoding apparatuses 100 and 200.

TABLE 1

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

[0217] 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 the coding units 1010, and the transformation units 1070 are transformation units of each of the coding units 1010.

[0218] 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.

[0219] In the prediction units 1060, some encoding units 1014, 1016, 1022, 1032, 1048, 1050, 1052, and 1054 are obtained by splitting the coding units in the encoding units 1010. 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.

[0220] 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, and 1052 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 and decoding apparatuses 100 and 200 may perform intra predic-

[0222] The output unit 130 of the video encoding apparatus 100 may output the encoding information about the coding units having a tree structure, and the image data and encoding information extractor 220 of the video decoding apparatus 200 may extract the encoding information about the coding units having a tree structure from a received bitstream.

[0223] 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, 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.

[0224] 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  $2N \times 2N$ .

[0225] The information about the partition type may indicate symmetrical partition types having sizes of  $2N \times 2N$ ,  $2N \times N$ ,  $N \times 2N$ , and  $N \times N$ , which are obtained by symmetrically splitting a height or a width of a prediction unit, and asymmetrical partition types having sizes of  $2N \times nU$ ,  $2N \times nD$ ,  $nL \times 2N$ , and  $nR \times 2N$ , which are obtained by asymmetrically splitting the height or width of the prediction unit. The asymmetrical partition types having the sizes of  $2N \times nU$  and  $2N \times nD$  may be respectively obtained by splitting the height of the prediction unit in 1:3 and 3:1, and the asymmetrical partition types having the sizes of  $nL \times 2N$  and  $nR \times 2N$  may be respectively obtained by splitting the width of the prediction unit in 1:3 and 3:1.



[0226] 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  $2N \times 2N$ , 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  $N/2 \times N/2$ .

[0227] 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.

[0228] 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.

[0229] 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.

[0230] 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 for predicting the current coding unit.

[0231] FIG. 21 is a diagram for describing a relationship between a coding unit, a prediction unit or a partition, and a transformation unit, according to encoding mode information of Table 1.

[0232] 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  $nR \times 2N$ .

[0233] Split information (TU size flag) of a transformation unit is a type of a transformation index, and a current size of a transformation unit may be determined based on the transformation index and a prediction unit type or a partition type of a current coding unit.

[0234] For example, 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 a TU size flag is 0, and a transformation unit 1344 having a size of  $N \times N$  is set if a TU size flag is 1.

[0235] On the other hand, 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.

[0236] Accordingly, the size of a transformation unit corresponding to the transformation index may vary according to a prediction unit type or a partition type of a coding unit.

[0237] Referring to FIG. 21, the TU size flag 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 having a tree structure while the TU size flag increases from 0.

[0238] In this case, the size of a transformation unit that has been actually used may be expressed by using a TU size flag of a transformation unit, according to an exemplary embodiment, together with a maximum size and minimum size of the transformation unit. According to an exemplary embodiment, the video encoding apparatus 100 is capable of encoding maximum transformation unit size information, minimum transformation unit size information, and a maximum TU size flag. The result of encoding the maximum transformation unit size information, the minimum transformation unit size information, and the maximum TU size flag may be inserted into an SPS. According to an exemplary embodiment, the video decoding apparatus 200 may decode video by using the maximum transformation unit size information, the minimum transformation unit size information, and the maximum TU size flag.

[0239] For example, if the size of a current coding unit is  $64 \times 64$  and a maximum transformation unit size is  $32 \times 32$ , then the size of a transformation unit may be  $32 \times 32$  when a TU size flag is 0, may be  $16 \times 16$  when the TU size flag is 1, and may be  $8 \times 8$  when the TU size flag is 2.

[0240] As another example, if the size of the current coding unit is  $32 \times 32$  and a minimum transformation unit size is  $32 \times 32$ , then the size of the transformation unit may be  $32 \times 32$  when the TU size flag is 0. Here, the TU size flag cannot be set to a value other than 0, since the size of the transformation unit cannot be less than  $32 \times 32$ .

[0241] As another example, if the size of the current coding unit is  $64 \times 64$  and a maximum TU size flag is 1, then the TU size flag may be 0 or 1. Here, the TU size flag cannot be set to a value other than 0 or 1.

[0242] Thus, if it is defined that the maximum TU size flag is 'MaxTransformSizeIndex', a minimum transformation unit size is 'MinTransformSize', and a transformation unit size is 'RootTuSize' when the TU size flag is 0, then a current minimum transformation unit size 'CurrMinTuSize' that can be determined in a current coding unit, may be defined by Equation (1):

$$\text{CurrMinTuSize} = \max(\text{MinTransformSize}, \text{RootTuSize} / (2^{\text{MaxTransformSizeIndex}})) \quad \text{Equation (1)}$$

[0243] Compared to the current minimum transformation unit size 'CurrMinTuSize' that can be determined in the current coding unit, a transformation unit size 'RootTuSize' when the TU size flag is 0 may denote a maximum transformation unit size that can be selected in the system. In Equation (1), 'RootTuSize/(2<sup>MaxTransformSizeIndex</sup>)' denotes a transformation unit size when the transformation unit size 'RootTuSize', when the TU size flag is 0, is split a number of times corresponding to the maximum TU size flag, and 'MinTransformSize' denotes a minimum transformation size. Thus, a smaller value from among 'RootTuSize/(2<sup>MaxTransformSizeIndex</sup>)' and 'MinTransformSize' may be the current minimum transformation unit size 'CurrMinTuSize' that can be determined in the current coding unit.

[0244] According to an exemplary embodiment, the maximum transformation unit size *RootTuSize* may vary according to the type of a prediction mode.

[0245] For example, if a current prediction mode is an inter mode, then '*RootTuSize*' may be determined by using Equation (2) below. In Equation (2), '*MaxTransformSize*' denotes a maximum transformation unit size, and '*PUSize*' denotes a current prediction unit size.

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

[0246] That is, if the current prediction mode is the inter mode, the transformation unit size '*RootTuSize*' when the TU size flag is 0, may be a smaller value from among the maximum transformation unit size and the current prediction unit size.

[0247] If a prediction mode of a current partition unit is an intra mode, '*RootTuSize*' may be determined by using Equation (3) below. In Equation (3), '*PartitionSize*' denotes the size of the current partition unit.

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

[0248] That is, if the current prediction mode is the intra mode, the transformation unit size '*RootTuSize*' when the TU size flag is 0 may be a smaller value from among the maximum transformation unit size and the size of the current partition unit.

[0249] However, the current maximum transformation unit size '*RootTuSize*' that varies according to the type of a prediction mode in a partition unit is just an example and is not limited thereto.

[0250] According to an exemplary embodiment, the current maximum transformation unit size '*RootTuSize*' may be equal to the size of the current coding unit. According to other exemplary embodiment, the current maximum transformation unit size '*RootTuSize*' may be determined base on a prediction unit type or a partition type of the current prediction unit or partition. For example, the current maximum transformation unit size '*RootTuSize*' may indicate the size of a maximum square included in the current prediction unit or partition.

[0251] FIG. 22 is a flowchart illustrating a video encoding method that uses a transformation index on the basis of coding units and transformation units having a tree structure, according to an exemplary embodiment.

[0252] In operation 1210, a current picture is split into at least one maximum coding unit. A maximum depth that indicates the total number of possible splits may be predetermined.

[0253] In operation 1220, a coded depth to output a final encoding result according to at least one split region, which is obtained by splitting a region of each maximum coding unit according to depths, is determined by encoding the at least one split region, and a coding unit according to a tree structure is determined.

[0254] The maximum coding unit is spatially split whenever the depth deepens, and thus is split into coding units of a lower depth. Each coding unit may be split into coding units of another lower depth by being spatially split independently from adjacent coding units. Encoding is repeatedly performed on each coding unit according to depths.

[0255] Also, a transformation unit according to partition types having the least encoding error is determined for each deeper coding unit. In order to determine a coded depth having a minimum encoding error in each maximum coding

unit, encoding errors may be measured and compared in all deeper coding units according to depths.

[0256] In the determination of the coding unit, transformation units, being a data unit in which the coding unit is transformed, may be determined. The transformation units may be determined to be a data unit minimizing an error due to the transformation on the coding unit. The transformation units may be determined to have the same size within a single coding unit. As a result of performing transformation at each level according to a transformation depth within a current coding unit, transformation units based on a tree structure that form a hierarchical structure between transformation units on the same region according to transformation depths and are independent from transformation units on the other region may be determined.

[0257] In operation 1230, encoded image data constituting the final encoding result according to the coded depth is output for each maximum coding unit, with encoding information about the coded depth and an encoding mode. The information about the encoding mode may include information that indicates a coded depth or split information, information that indicates a partition type of a prediction unit, information that indicates a prediction mode, information that indicates a size of a transformation unit, and a transformation index. The encoded information about the encoding mode may be transmitted to a decoder with the encoded image data.

[0258] FIG. 23 is a flowchart illustrating a video decoding method that use a transformation index on the basis of the coding units and the transformation units having a tree structure, according to an exemplary embodiment.

[0259] In operation 1310, a bitstream of an encoded video is received and parsed.

[0260] In operation 1320, encoded image data of a current picture assigned to a maximum coding unit, and information about a coded depth and an encoding mode according to maximum coding units are extracted from the parsed bitstream. The coded depth of each maximum coding unit is a depth having the least encoding error in each maximum coding unit. In encoding each maximum coding unit, the image data is encoded based on at least one data unit obtained by hierarchically splitting the each maximum coding unit according to depths.

[0261] According to the information that indicates the coded depth and the encoding mode, the maximum coding unit may be split into coding units having a tree structure. Each of the coding units having the tree structure is determined as a coding unit corresponding to a coded depth, and is optimally encoded as to output the least encoding error. Accordingly, encoding and decoding efficiency of an image may be improved by decoding each piece of encoded image data in the coding units after determining at least one coded depth according to coding units.

[0262] According to the transformation index included in the information about the encoding mode, transformation units having a tree structure within a coding unit may be determined. For example, the number of splits from the current coding unit to a transformation unit may be read from the transformation index. In another embodiment, it may be determined whether the current coding unit is split into transformation units of a lower level, and thus a structure of transformation units having a tree structure may be finally read from a bitstring that indicates whether splitting from the

uppermost transformation unit to a lower transformation unit is performed for each region of the current coding unit.

[0263] In operation 1330, the image data of each maximum coding unit is decoded based on the information about the coded depth and the encoding mode according to the maximum coding units. The decoded image data may be reproduced by a reproducing apparatus, stored in a storage medium, or transmitted through a network.

[0264] The exemplary embodiments can be written as computer programs and can be implemented in general-use digital computers that execute the programs using a computer readable recording medium. Examples of the computer readable recording medium include magnetic storage media (e.g., ROM, floppy disks, hard disks, etc.) and optical recording media (e.g., CD-ROMs, or DVDs). Alternatively, the exemplary embodiments may be embodied as signals computer-readable transmission media, such as data signals, for transmission over a computer network, for example the Internet.

[0265] The video encoding apparatuses or video decoding apparatuses of the exemplary embodiments may include a bus coupled to every unit of the apparatus, at least one processor connected to the bus that executes commands, and a memory connected to the bus that stores commands, received messages, and generated messages.

[0266] 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. The exemplary embodiments should be considered in descriptive sense only and not for purposes of limitation. 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 an encoded video comprising:

receiving a bitstream of an encoded video;

determining at least one coding unit by using split information extracted from the bitstream;

extracting from the bitstream transformation index information indicating whether a transformation unit of a current level included in a coding unit among the at least one coding unit is split;

when the transformation index information indicates a split of the transformation unit of the current level, splitting the transformation unit of the current level into transformation units of a lower level; and

when the transformation index information indicates a non-split of the transformation unit of the current level, performing an inverse transformation on the transformation unit of the current level to generate residual data corresponding to the transformation unit of the current level,

wherein the transformation unit of the current level is split into four transformation units of the lower level.

2. The video decoding method of claim 1, wherein the transformation unit of the current level is included in the coding unit, and a size of the transformation unit of the current level is smaller than or equal to a size of the coding unit.

3. The video decoding method of claim 2, wherein the transformation unit of the current level is obtained by halving a height and a width of the coding unit.

4. The video decoding method of claim 1, wherein the coding unit is a data unit in which a picture of the encoded video is encoded and the transformation unit of the current level is a data unit in which the data of the coding unit is transformed.

\* \* \* \* \*