



US012395638B2

(12) **United States Patent**  
**Lim et al.**

(10) **Patent No.:** **US 12,395,638 B2**  
(45) **Date of Patent:** **Aug. 19, 2025**

(54) **IMAGE ENCODING/DECODING METHOD AND DEVICE, AND RECORDING MEDIUM FOR STORING BITSTREAM**

(71) Applicant: **Electronics and Telecommunications Research Institute**, Daejeon (KR)

(72) Inventors: **Woong Lim**, Daejeon (KR); **Gun Bang**, Daejeon (KR); **Hui Yong Kim**, Daejeon (KR)

(73) Assignee: **Electronics and Telecommunications Research Institute**, Daejeon (KR)

(\*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 4 days.

(21) Appl. No.: **17/613,405**

(22) PCT Filed: **Jun. 19, 2020**

(86) PCT No.: **PCT/KR2020/007973**

§ 371 (c)(1),

(2) Date: **Nov. 22, 2021**

(87) PCT Pub. No.: **WO2020/256470**

PCT Pub. Date: **Dec. 24, 2020**

(65) **Prior Publication Data**

US 2022/0248029 A1 Aug. 4, 2022

(30) **Foreign Application Priority Data**

Jun. 19, 2019 (KR) ..... 10-2019-0072909

Jun. 24, 2019 (KR) ..... 10-2019-0075129

(Continued)

(51) **Int. Cl.**

**H04N 19/23** (2014.01)

**H04N 19/14** (2014.01)

(Continued)

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

CPC ..... **H04N 19/14** (2014.11); **H04N 19/174** (2014.11); **H04N 19/176** (2014.11); **H04N 19/1883** (2014.11)

(58) **Field of Classification Search**

CPC .... **H04N 19/14**; **H04N 19/174**; **H04N 19/176**; **H04N 19/1883**; **H04N 19/172**;  
(Continued)

(56) **References Cited**

U.S. PATENT DOCUMENTS

9,148,663 B2 \* 9/2015 Lee ..... **H04N 19/82**

9,204,148 B1 \* 12/2015 Lee ..... **H04N 19/117**

(Continued)

FOREIGN PATENT DOCUMENTS

CN 101998141 A 3/2011

GB 2584723 \* 6/2019 ..... **H04N 19/119**

(Continued)

OTHER PUBLICATIONS

Benjamin Bross, Versatile Video Coding (Draft 5) (Year: 2019).\*

(Continued)

Primary Examiner — Philip P. Dang

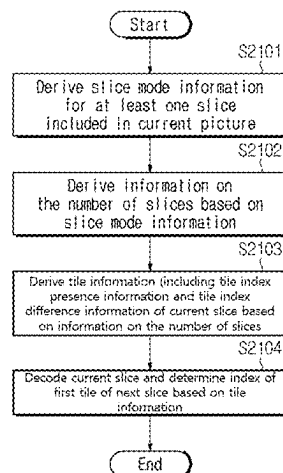
(74) Attorney, Agent, or Firm — BROADVIEW IP LAW, PC

(57)

**ABSTRACT**

Disclosed herein are an image encoding/decoding method and apparatus, and a recording medium storing a bitstream. The image decoding method according to the present invention includes: deriving slice mode information for at least one slice comprised in a current picture; deriving information on the number of slices based on the slice mode information; deriving tile information of a current slice based on the information on the number of slices; and, decoding the current slice based on the tile information. Herein, the tile information includes tile index presence information and tile index difference value information, and

(Continued)



the tile index difference value information is not signaled for a last slice included in the current picture.

### 9 Claims, 19 Drawing Sheets

#### (30) Foreign Application Priority Data

Oct. 7, 2019 (KR) ..... 10-2019-0123544  
Oct. 10, 2019 (KR) ..... 10-2019-0125022

#### (51) Int. Cl.

**H04N 19/169** (2014.01)  
**H04N 19/174** (2014.01)  
**H04N 19/176** (2014.01)

#### (58) Field of Classification Search

CPC .... H04N 19/105; H04N 19/129; H04N 19/61;  
H04N 19/96; H04N 19/593; H04N  
19/119; H04N 19/503; H04N 19/70;  
H04N 19/132; H04N 19/52; H04N 19/50  
See application file for complete search history.

#### (56) References Cited

##### U.S. PATENT DOCUMENTS

9,204,171 B1 \* 12/2015 Lee ..... H04N 19/117  
9,270,990 B2 \* 2/2016 Lee ..... H04N 19/82  
10,477,247 B2 \* 11/2019 Kang ..... H04N 19/186  
10,911,756 B2 \* 2/2021 Lee ..... H04N 19/176  
10,958,903 B2 \* 3/2021 Jun ..... H04N 19/159  
11,039,130 B2 \* 6/2021 Lee ..... H04N 19/593  
11,102,509 B2 \* 8/2021 Ko ..... H04N 19/11  
11,166,013 B2 \* 11/2021 Hannuksela ..... H04N 19/593  
11,290,728 B2 \* 3/2022 Hannuksela ..... H04N 19/70  
11,336,899 B2 \* 5/2022 Jun ..... H04N 19/159  
11,368,681 B2 \* 6/2022 Lee ..... H04N 19/107  
11,405,620 B2 \* 8/2022 Jun ..... H04N 19/126  
11,451,767 B2 \* 9/2022 Jun ..... H04N 19/11  
11,509,930 B2 \* 11/2022 Lim ..... H04N 19/176

11,722,694 B2 \* 8/2023 Galpin ..... H04N 19/157  
375/240.12  
2014/0192899 A1 \* 7/2014 Wang ..... H04N 19/44  
375/240.26  
2019/0037216 A1 \* 1/2019 Cho ..... H04N 19/159  
2019/0068989 A1 \* 2/2019 Lee ..... H04N 19/513  
2019/0082178 A1 3/2019 Kim et al.  
2020/0084441 A1 \* 3/2020 Lee ..... H04N 19/139

##### FOREIGN PATENT DOCUMENTS

JP 2018-201234 A 12/2018  
KR 1020130043054 A 4/2013  
KR 1020180101123 A 9/2018  
KR 1020190018938 A 2/2019  
KR 1020190039790 A 4/2019  
WO 2012/033327 A2 3/2012  
WO 2018/182185 A1 10/2018

##### OTHER PUBLICATIONS

JVET-M0121 (Year: 2019).  
JVET-N0496 (Year: 2019).  
Benjamin Bross et al., Versatile Video Coding (Draft 4), JVET-M1001-v7, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 13th Meeting: Marrakech, MA, Jan. 9-18, 2019.  
Benjamin Bross et al., Versatile Video Coding (Draft 5), JVET-N1001-v7, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 14th Meeting: Geneva, CH, Mar. 19-27, 2019.  
Rickard Sjöberg et al., AHG12: Rectangular tile group address signaling, JVET-N0496-v3, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 14th Meeting: Geneva, CH, Mar. 19-27, 2019.  
Yong He et al., AHG12: On Rectangular Tile Group, JVET-M0121r1, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 13th Meeting: Marrakech, MA, Jan. 9-18, 2019, InterDigital Communications, Inc.  
Liu, Shaoli et al. (2014) "Auxiliary stream for optimizing memory access of video decoders," Science China. doi: 10.1007/s11432-013-5045-9.

\* cited by examiner

FIG. 1

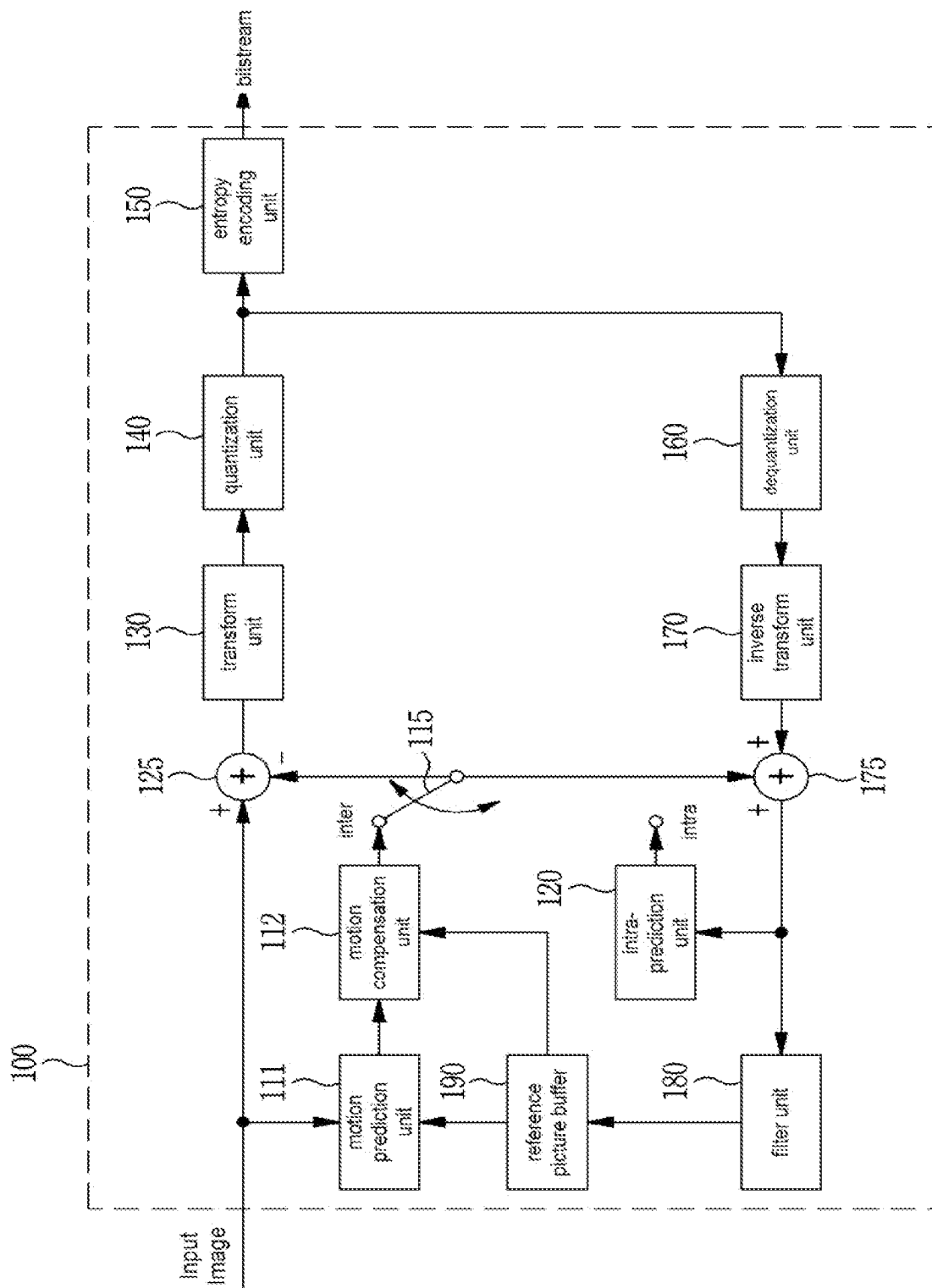


FIG. 2

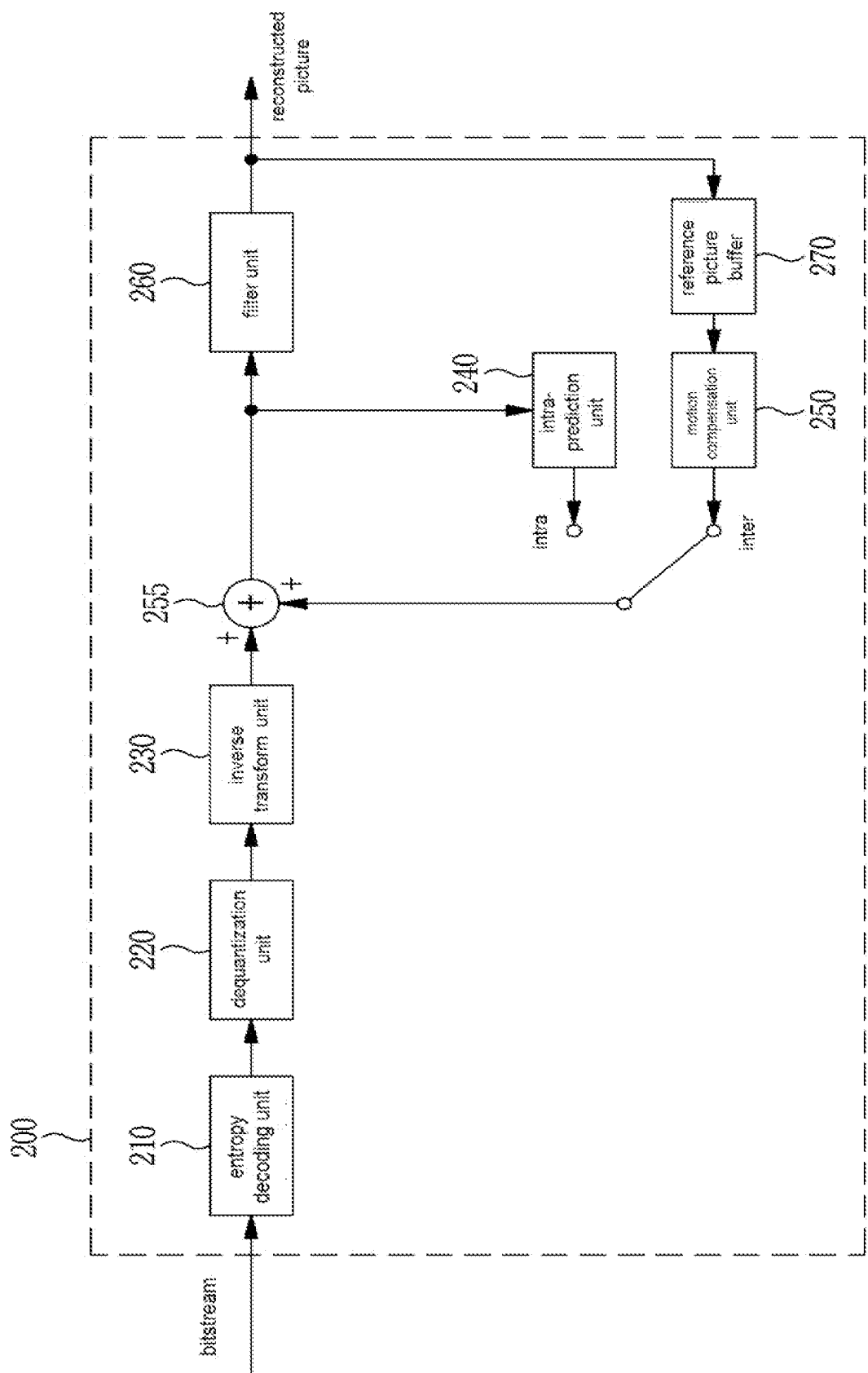


FIG. 3

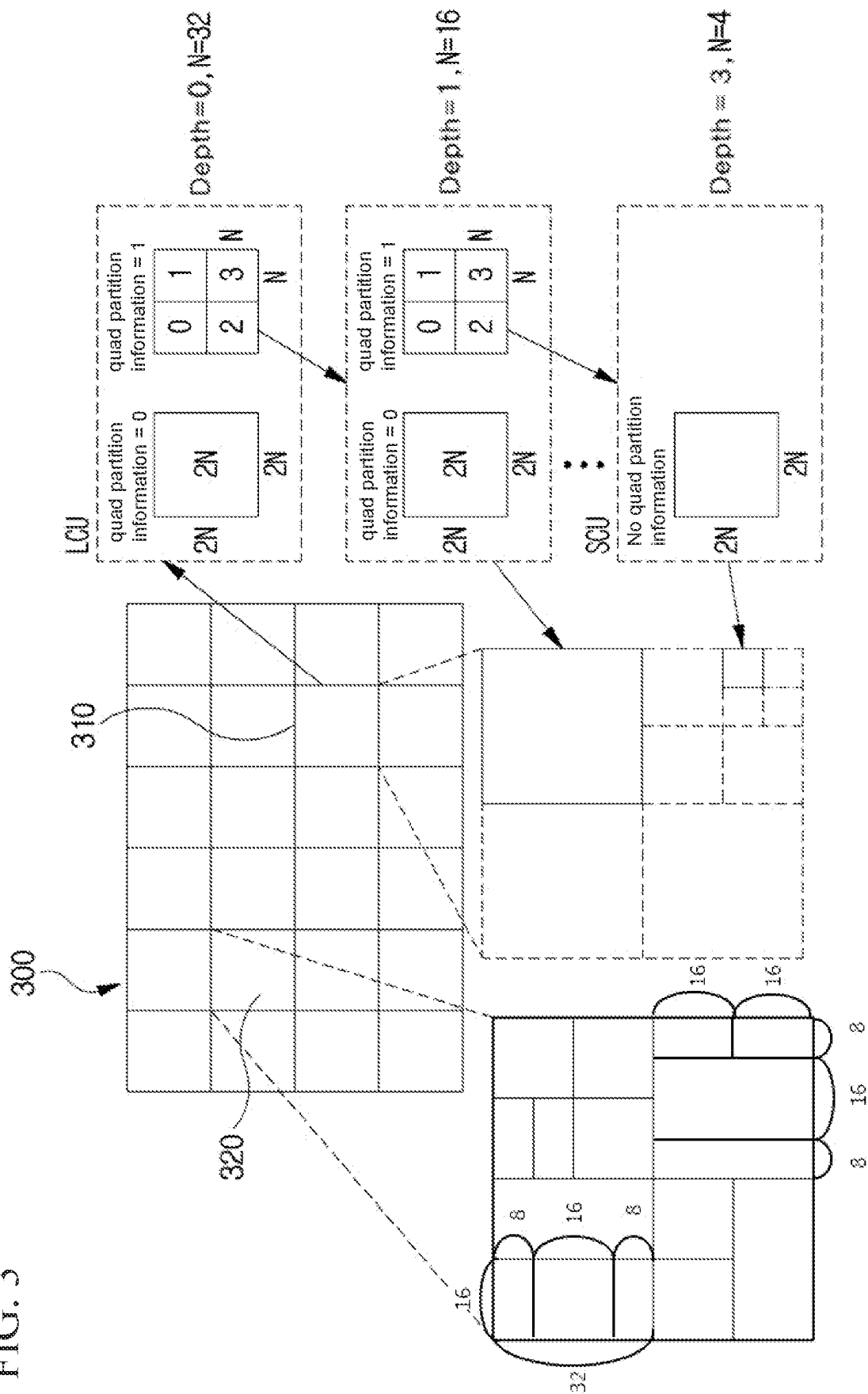


FIG. 4

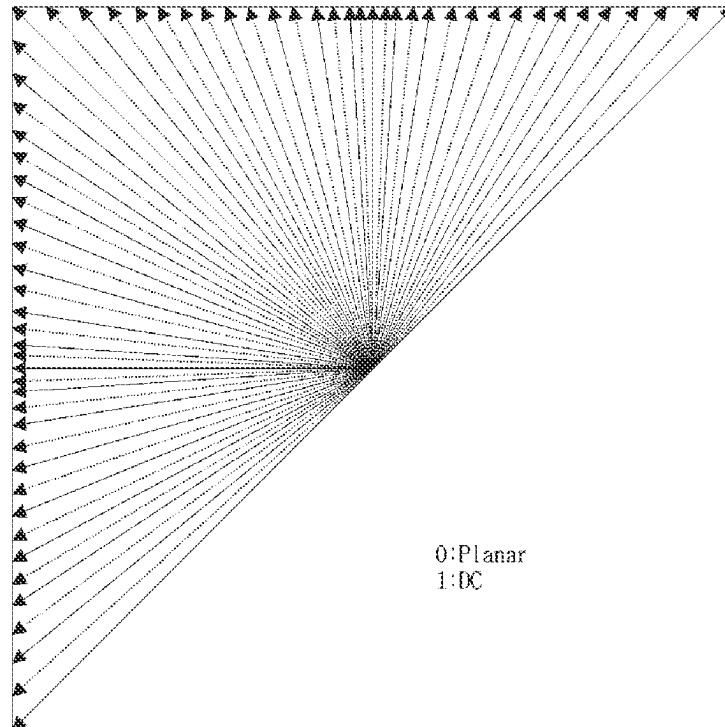


FIG. 5

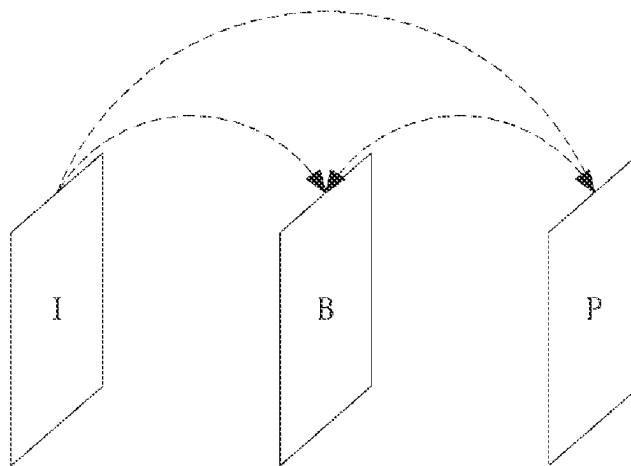


FIG. 6

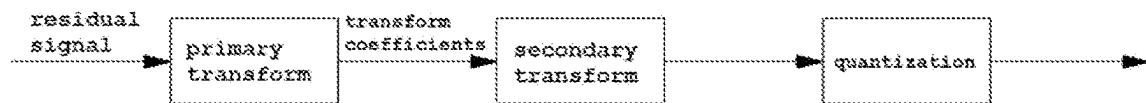


FIG. 7

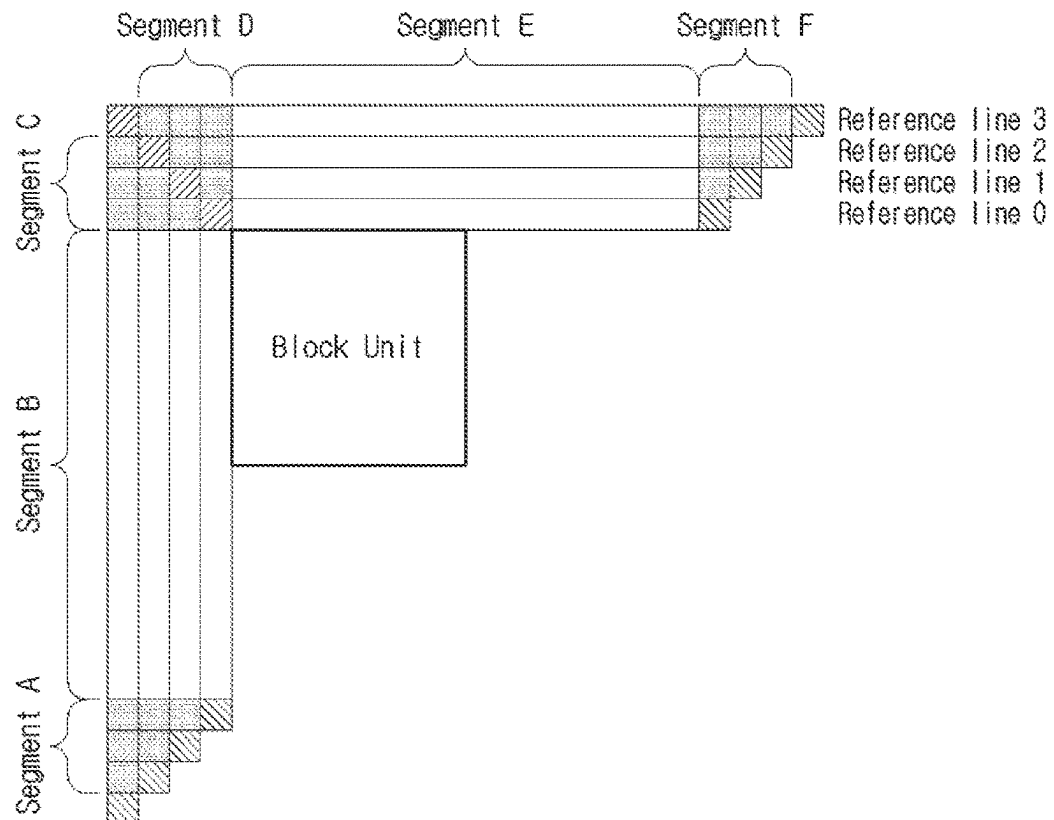


FIG. 8

pic_parameter_set_rbsp() {	Descriptor
...	
single_brick_per_slice_flag	u(1)
if( !single_brick_per_slice_flag )	
rect_slice_flag	u(1)
if( rect_slice_flag && !single_brick_per_slice_flag ) {	
num_slices_in_pic_minus1	ue(v)
for( i=0; i <= num_slices_in_pic_minus1; i++ ) {	
if( i > 0 )	
top_left_brick_ids[ i ]	u(v)
bottom_right_brick_ids_delta[ i ]	u(v)
}	
}	
...	
}	



## FIG. 9

`bottom_right_brick_idx_delta[ i ]` specifies the difference between the brick index of the brick located at the bottom-right corner of the  $i$ -th slice and `top_left_brick_idx[ i ]`. When `single_brick_per_slice_flag` is equal to 1, the value of `bottom_right_brick_idx_delta[ i ]` is inferred to be equal to 0. The length of the `bottom_right_brick_idx_delta[ i ]` syntax element is  $\text{Ceil}(\text{Log2}(\text{NumBricksInPic} - \text{top\_left\_brick\_idx}[ i ]))$  bits.

It is a requirement of bitstream conformance that a slice shall include either a number of complete tiles or only a consecutive sequence of complete bricks of one tile.

The variable `NumBricksInSlice[ i ]` and `BricksToSliceMap[ j ]`, which specify the number of bricks in the  $i$ -th slice and the mapping of bricks to slices, are derived as follows:

```

NumBricksInSlice[ i ] = 0
botRightBkIdx = top_left_brick_idx[ i ] + bottom_right_brick_idx_delta[ i ]
for( j = 0; j < NumBricksInPic; j++ ) {
    if( BrickColBd[ j ] >= BrickColBd[ top_left_brick_idx[ i ] ] &&

BrickColBd[ j ] <= BrickColBd[ botRightBkIdx ] &&

BrickRowBd[ j ] >= BrickRowBd[ top_left_brick_idx[ i ] ] &&                                     (7-35)

BrickRowBd[ j ] <= BrickRowBd[ botRightBkIdx ] ) {
    NumBricksInSlice[ i ]++
    BricksToSliceMap[ j ] = i
}
}

```

FIG. 10

pic_parameter_set_rbsp() {	Descriptor
...	
single_brick_per_slice_flag	u(1)
if( !single_brick_per_slice_flag )	
rect_slice_flag	u(1)
if( rect_slice_flag && !single_brick_per_slice_flag ) {	
num_slices_in_pic_minus1	ue(v)
for( i=0; i <= num_slices_in_pic_minus1; i++ ) {	
if( i > 0 )	
top_left_brick_idx[ i ]	u(v)
if( i != num_slices_in_pic_minus1 )	
bottom_right_brick_idx_delta[ i ]	u(v)
}	
}	
...	
}	

## FIG. 11

`bottom_right_brick_idx_delta[ i ]` specifies the difference between the brick index of the brick located at the bottom-right corner of the  $i$ -th slice and `top_left_brick_idx[ i ]`. When `single_brick_per_slice_flag` is equal to 1, the value of `bottom_right_brick_idx_delta[ i ]` is inferred to be equal to 0. The length of the `bottom_right_brick_idx_delta[ i ]` syntax element is  $\text{Ceil}(\text{Log2}(\text{NumBricksInPic} - \text{top\_left\_brick\_idx}[ i ]))$  bits.

It is a requirement of bitstream conformance that a slice shall include either a number of complete tiles or only a consecutive sequence of complete bricks of one tile.

The variable `NumBricksInSlice[ i ]` and `BricksToSliceMap[ j ]`, which specify the number of bricks in the  $i$ -th slice and the mapping of bricks to slices, are derived as follows:

```

NumBricksInSlice[ i ] = 0
if( i != num_slices_in_pic_minus1 )
    botRightBkIdx = top_left_brick_idx[ i ] + bottom_right_brick_idx_delta[ i ]
else
    botRightBkIdx = NumBricksInPic - 1;
for( j = 0; j < NumBricksInPic; j++ ) {
    if( BrickColBd[ j ] >= BrickColBd[ top_left_brick_idx[ i ] ] &&

BrickColBd[ j ] <= BrickColBd[ botRightBkIdx ] &&

BrickRowBd[ j ] >= BrickRowBd[ top_left_brick_idx[ i ] ] &&                                     (7-35)

BrickRowBd[ j ] <= BrickRowBd[ botRightBkIdx ] ) {
    NumBricksInSlice[ i ]++
    BricksToSliceMap[ j ] = i
    }
}

```

FIG. 12

pic_parameter_set_rbsp() {	Descriptor
...	
brick_splitting_present_flag	u(1)
for( i=0; brick_splitting_present_flag && i < NumTilesInPic; i++ ) {	
brick_split_flag[ i ]	u(1)
if( brick_split_flag[ i ] ) {	
uniform_brick_spacing_flag[ i ]	u(1)
if( uniform_brick_spacing_flag[ i ] )	
brick_height_minus1[ i ]	ue(v)
else {	
num_brick_rows_minus1[ i ]	ue(v)
for( j=0; j < num_brick_rows_minus1[ i ]; j++ )	
brick_row_height_minus1[ i ][ j ]	ue(v)
}	
}	
}	
...	
}	

FIG. 13

```
for ( brickIdx = 0, i = 0; i < NumTilesInPic; i++ ) {  
    tileX = i % ( num_tile_columns_minus1 + 1 )  
    tileY = i / ( num_tile_columns_minus1 + 1 )  
    if( !brick_split_flag[ i ] ) {  
        BrickColBd[ brickIdx ] = tileColBd[ tileX ]  
        BrickRowBd[ brickIdx ] = tileRowBd[ tileY ]  
        BrickWidth[ brickIdx ] = colWidth[ tileX ]  
        BrickHeight[ brickIdx ] = RowHeight[ tileY ]  
        brickIdx++  
    } else {  
        if( uniform_brick_spacing_flag[ i ] ) {  
            remainingHeightInCtbsY = RowHeight[ tileY ]  
            j = 0  
            while( remainingHeightInCtbsY > ( brick_height_minus1[ i ] + 1 ) ) {  
                rowHeight2[ j++ ] = brick_height_minus1[ i ] + 1  
                remainingHeightInCtbsY -= ( brick_height_minus1[ i ] + 1 )  
            }  
            rowHeight2[ j ] = remainingHeightInCtbsY  
            num_brick_rows_minus1[ i ] = j  
        }  
    }  
}
```

FIG. 14

```
else {  
    rowHeight2[ num_brick_rows_minus1[ i ] ] = RowHeight[ tileY ]  
    for( j = 0; j < num_brick_rows_minus1[ i ]; j++ ) {  
        rowHeight2[ j ] = brick_row_height_minus1[ i ][ j ] + 1  
        rowHeight2[ num_brick_rows_minus1[ i ] ] -= rowHeight2[ j ]  
    }  
}  
for( rowBd2[ 0 ] = 0, j = 0; j <= num_brick_rows_minus1[ i ]; j++ )  
    rowBd2[ j + 1 ] = rowBd2[ j ] + rowHeight2[ j ]  
for( j = 0; j < num_brick_rows_minus1[ i ] + 1; j++ ) {  
    BrickColBd[ brickIdx ] = tileColBd[ tileX ]  
    BrickRowBd[ brickIdx ] = tileRowBd[ tileY ] + rowBd2[ j ]  
    BrickWidth[ brickIdx ] = colWidth[ tileX ]  
    BrickHeight[ brickIdx ] = rowHeight2[ j ]  
    brickIdx ++  
}  
}  
NumBricksInPic = brickIdx
```

FIG. 15

pic_parameter_set_rbsp() {	Descriptor
...	
brick_splitting_present_flag	u(1)
for( i=0; brick_splitting_present_flag && i < NumTilesInPic; i++ ) {	
brick_split_flag[ i ]	u(1)
if( brick_split_flag[ i ] ) {	
uniform_brick_spacing_flag[ i ]	u(1)
if( uniform_brick_spacing_flag[ i ] )	
brick_height_minus1[ i ]	ue(v)
else {	
num_brick_rows_minus2[ i ]	ue(v)
for( j=0; j < num_brick_rows_minus2[ i ]; j++ )	
brick_row_height_minus1[ i ][ j ]	ue(v)
}	
}	
}	
...	
}	

FIG. 16

```
for ( brickIdx = 0, i = 0; i < NumTilesInPic; i++ ) {  
    tileX = i % ( num_tile_columns_minus1 + 1 )  
    tileY = i / ( num_tile_columns_minus1 + 1 )  
    if( !brick_split_flag[ i ] ) {  
        BrickColBd[ brickIdx ] = tileColBd[ tileX ]  
        BrickRowBd[ brickIdx ] = tileRowBd[ tileY ]  
        BrickWidth[ brickIdx ] = colWidth[ tileX ]  
        BrickHeight[ brickIdx ] = RowHeight[ tileY ]  
        brickIdx++  
    } else {  
        if( uniform_brick_spacing_flag[ i ] ) {  
            remainingHeightInCtbsY = RowHeight[ tileY ]  
            j = 0  
            while( remainingHeightInCtbsY > ( brick_height_minus1[ i ] + 1 ) ) {  
                rowHeight2[ j++ ] = brick_height_minus1[ i ] + 1  
                remainingHeightInCtbsY -= ( brick_height_minus1[ i ] + 1 )  
            }  
            rowHeight2[ j ] = remainingHeightInCtbsY  
            num_brick_rows_minus2[ i ] = j - 1  
        }  
    }  
}
```



FIG. 17

```
else {
    rowHeight2[ num_brick_rows_minus2[ i ] + 1 ] = RowHeight[ tileY ]
    for( j = 0; j < num_brick_rows_minus2[ i ] + 1; j++ ) {
        rowHeight2[ j ] = brick_row_height_minus1[ i ][ j ] + 1
        rowHeight2[ num_brick_rows_minus2[ i ] + 1 ] -= rowHeight2[ j ]
    }
}
for( rowBd2[ 0 ] = 0, j = 0; j <= num_brick_rows_minus2[ i ] + 1; j++ )
    rowBd2[ j + 1 ] = rowBd2[ j ] + rowHeight2[ j ]
for( j = 0; j < num_brick_rows_minus2[ i ] + 2; j++ ) {
    BrickColBd[ brickIdx ] = tileColBd[ tileX ]
    BrickRowBd[ brickIdx ] = tileRowBd[ tileY ] + rowBd2[ j ]
    BrickWidth[ brickIdx ] = colWidth[ tileX ]
    BrickHeight[ brickIdx ] = rowHeight2[ j ]
    brickIdx ++
}
}
}
NumBricksInPic = brickIdx
```

FIG. 18

num_slices_in_pic_minus1	uc(v)
for( i = 0; i <= num_slices_in_pic_minus1; i++ ) {	
slice_width_in_tiles_minus1[ i ]	uc(v)
slice_height_in_tiles_minus1[ i ]	uc(v)
if( slice_width_in_tiles_minus1[ i ] == 0 && slice_height_in_tiles_minus1[ i ] == 0 ) {	
num_slices_in_tile_minus1[ i ]	uc(v)
numSlicesInTileMinus1 = num_slices_in_tile_minus1[ i ]	
for( j = 0; j < numSlicesInTileMinus1; j++ )	
slice_height_in_ctu_minus1[ i++ ]	uc(v)
}	
if( i < num_slices_in_pic_minus1 )	
tile_idx_delta[ i ]	se(v)
}	

FIG. 19

num_slices_in_pic_minus1	ue(v)
for( i = 0; i <= num_slices_in_pic_minus1; i++ ) {	
if( i != 0 ) tile_idx_delta[ i ]	se(v)
slice_width_in_tiles_minus1[ i ]	ue(v)
slice_height_in_tiles_minus1[ i ]	ue(v)
if( slice_width_in_tiles_minus1[ i ] == 0 && slice_height_in_tiles_minus1[ i ] == 0 ) {	
num_slices_in_tile_minus1[ i ]	ue(v)
numSlicesInTileMinus1 = num_slices_in_tile_minus1[ i ]	
for( j = 0; j < numSlicesInTileMinus1; j++ )	
slice_height_in_ctu_minus1[ i++ ]	ue(v)
}	
}	

FIG. 20

num_slices_in_pic_minus1	ue(v)
for( i = 0; i < num_slices_in_pic_minus1; i++ ) {	
slice_width_in_tiles_minus1[ i ]	ue(v)
slice_height_in_tiles_minus1[ i ]	ue(v)
if( slice_width_in_tiles_minus1[ i ] == 0 && slice_height_in_tiles_minus1[ i ] == 0 ) {	
num_slices_in_tile_minus1[ i ]	ue(v)
numSlicesInTileMinus1 = num_slices_in_tile_minus1[ i ]	
for( j = 0; j < numSlicesInTileMinus1; j++ )	
slice_height_in_ctu_minus1[ i++ ]	ue(v)
}	
if( i < num_slices_in_pic_minus1 )	
tile_idx_delta[ i ]	se(v)
}	

FIG. 21

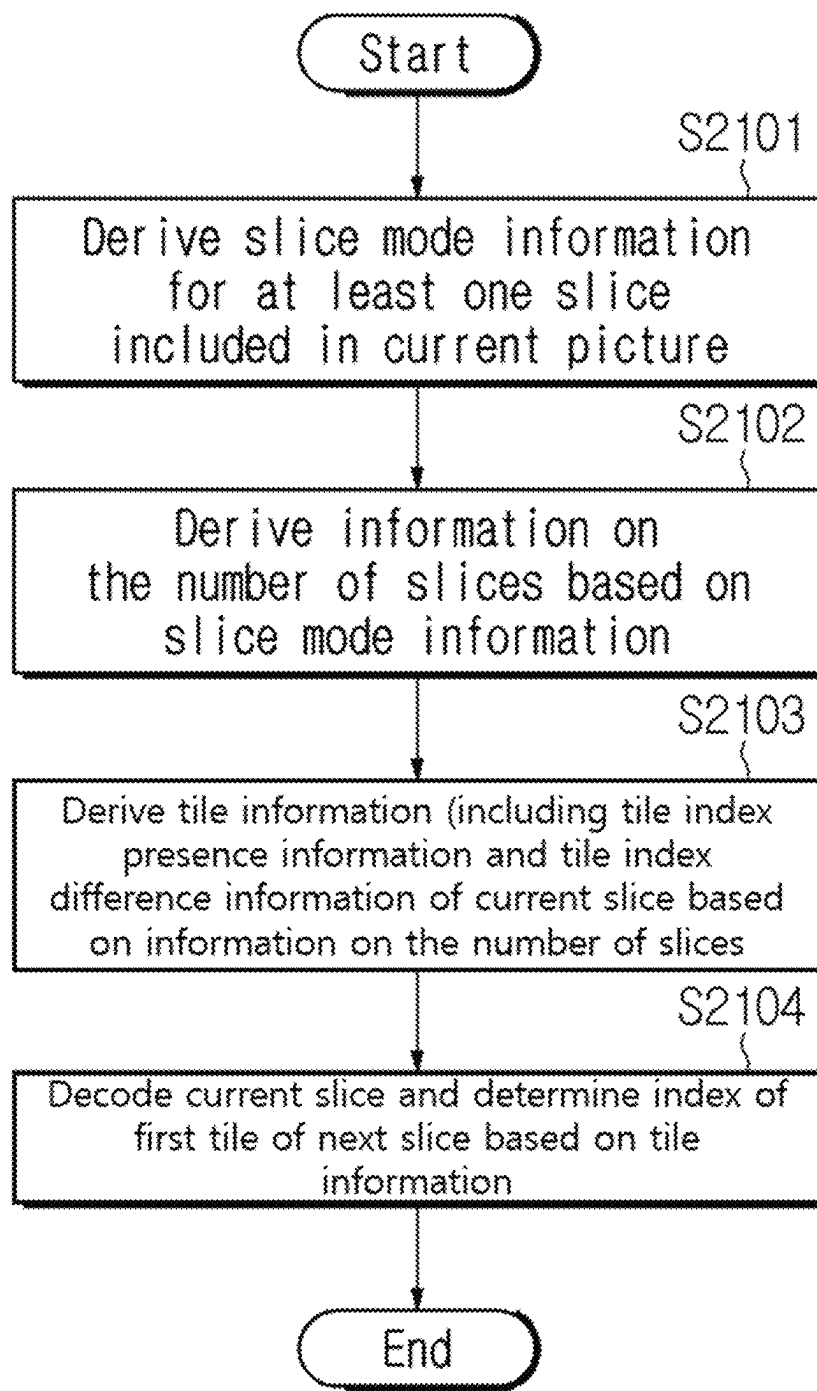
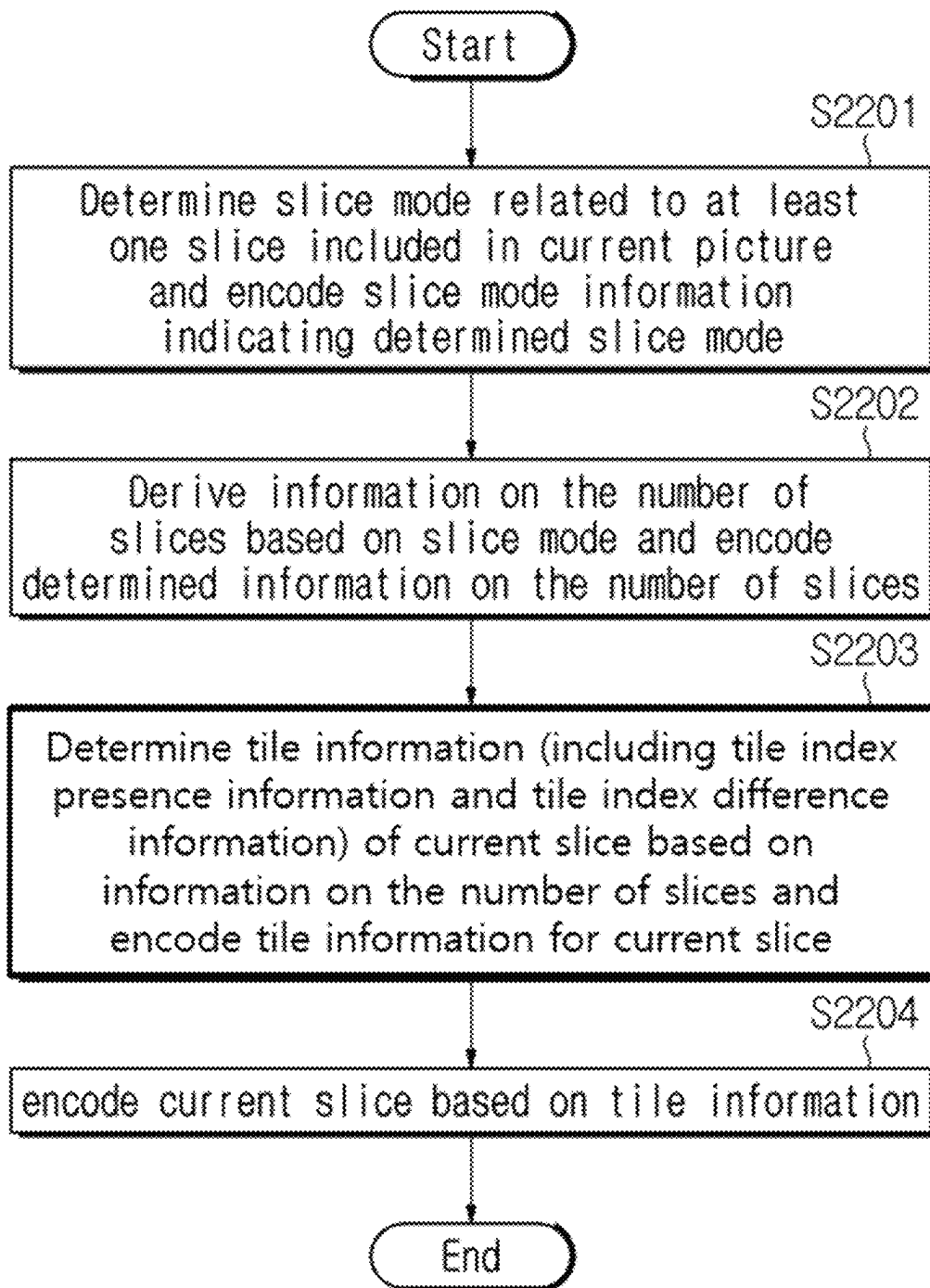


FIG. 22



1

# IMAGE ENCODING/DECODING METHOD AND DEVICE, AND RECORDING MEDIUM FOR STORING BITSTREAM

## TECHNICAL FIELD

The present invention relates to an image encoding/decoding method and apparatus, and a recording medium storing a bitstream. More particularly, the present invention relates to a transmitting method and apparatus for a rectangular region constituting an image screen and to a recording medium storing a bitstream.

## BACKGROUND ART

Recently, the demand for high resolution and quality images such as high definition (HD) or ultra-high definition (UHD) images has increased in various applications. As the resolution and quality of images are improved, the amount of data correspondingly increases. This is one of the causes of increase in transmission cost and storage cost when transmitting image data through existing transmission media such as wired or wireless broadband channels or when storing image data. In order to solve such problems with high resolution and quality image data, a high efficiency image encoding/decoding technique is required.

There are various video compression techniques such as an inter prediction technique of predicting the values of pixels within a current picture from the values of pixels within a preceding picture or a subsequent picture, an intra prediction technique of predicting the values of pixels within a region of a current picture from the values of pixels within another region of the current picture, a transform and quantization technique of compressing the energy of a residual signal, and an entropy coding technique of allocating frequently occurring pixel values with shorter codes and less occurring pixel values with longer codes.

## DISCLOSURE

### Technical Problem

The present invention is directed to provide a method and apparatus for encoding and decoding an image, which have enhanced encoding and decoding efficiency.

Also, the present invention is directed to provide a method and apparatus for efficiently transmitting a rectangular region constituting an image screen.

Also, the present invention is directed to provide a recording medium storing a bitstream generated by an image decoding method or apparatus according to the present invention.

### Technical Solution

An image decoding method according to an embodiment of the present invention comprising, deriving slice mode information for at least one slice comprised in a current picture; deriving information on the number of slices based on the slice mode information; deriving tile information of a current slice based on the information on the number of slices; and decoding the current slice based on the tile information, wherein the tile information comprises tile index presence information and tile index difference value information, and wherein the tile index difference value information is not signaled for a last slice comprised in the current picture.

2

In the image decoding method, wherein the slice mode information indicates whether at least one slice comprised in the current picture is in a first slice mode consisting of tiles according to a raster scan order or in a second slice mode consisting of a rectangular region.

In the image decoding method, wherein the slice mode information, the information on the number of slices, and the tile information are signaled through a picture parameter set.

In the image decoding method, wherein the deriving of the information on the number of slices, when the slice mode information is a second slice mode, comprises deriving the information on the number of slices related to at least one slice comprised in the current picture.

In the image decoding method, wherein the tile index presence information indicates whether or not at least one slice comprised in the current picture is specified by the tile index difference value information.

In the image decoding method, wherein the tile index difference value information indicates a difference value between a tile index of a slice comprising a first coding tree unit (CTU) in the current slice and a tile index of a slice comprising a CTU in a next slice of the current slice.

An image encoding method according to an embodiment of the present invention comprising, determining a slice mode related to at least one slice comprised in a current picture and encoding slice mode information indicating the determined slice mode; determining information on the number of slices based on the slice mode and encoding the determined information on the number of slices; determining tile information for a current slice based on the information on the number of slices and encoding the tile information for the current slice; and encoding the current slice based on the tile information, wherein the tile information comprises tile index presence information and tile index difference value information, and wherein the tile index difference value information is not signaled for a last slice comprised in the current picture.

In the image encoding method, wherein the slice mode information indicates whether at least one slice comprised in the current picture is in a first slice mode consisting of tiles according to a raster scan order or in a second slice mode consisting of a rectangular region.

In the image encoding method, wherein the slice mode information, the information on the number of slices, and the tile information are signaled through a picture parameter set.

In the image encoding method, wherein the determining of the information on the number of slices and encoding of the determined information on the number of slices, when the slice mode information is a second slice mode, comprises determining the information on the number of slices related to at least one slice comprised in the current picture.

In the image encoding method, wherein the tile index presence information indicates whether or not at least one slice comprised in the current picture is specified by the tile index difference value information.

In the image encoding method, wherein the tile index difference value information indicates a difference value between a tile index of a slice comprising a first coding tree unit (CTU) in the current slice and a tile index of a slice comprising a CTU in a next slice of the current slice.

A non-transitory computer-readable recording medium for storing a bitstream generated by an image encoding method, according to an embodiment of the present invention, the image encoding method comprising, determining a slice mode related to at least one slice comprised in a current picture and encoding slice mode information indicating the determined slice mode; determining information on the

number of slices based on the slice mode and encoding the determined information on the number of slices; determining tile information for a current slice based on the information on the number of slices and encoding the tile information for the current slice; and encoding the current slice based on the tile information, wherein the tile information comprises tile index presence information and tile index difference value information, and wherein the tile index difference value information is not signaled for a last slice comprised in the current picture.

#### Advantageous Effects

According to the present invention, it is possible to provide an image encoding/decoding method and apparatus with improved compression efficiency.

Also, according to the present invention, a method and apparatus for efficiently transmitting a rectangular region constituting an image screen may be provided.

Also, according to the present invention, a recording medium storing a bitstream generated by an image decoding method or apparatus of the present invention may be provided.

Also, according to the present invention, a recording medium storing a bitstream, which is received and decoded by an image decoding apparatus of the present invention and is used to reconstruct an image, may be provided.

#### DESCRIPTION OF DRAWINGS

FIG. 1 is a block diagram showing a configuration of an encoding apparatus according to an embodiment to which the present invention is applied.

FIG. 2 is a block diagram showing a configuration of a decoding apparatus according to an embodiment and to which the present invention is applied.

FIG. 3 is a view schematically showing a partition structure of an image when encoding and decoding the image.

FIG. 4 is a view showing an intra-prediction process.

FIG. 5 is a diagram illustrating an embodiment of an inter-picture prediction process.

FIG. 6 is a diagram illustrating a transform and quantization process.

FIG. 7 is a diagram illustrating reference samples capable of being used for intra prediction.

FIG. 8 is a view illustrating an example of signaling brick-related information according to an embodiment of the present invention.

FIG. 9 is a view for explaining an example of brick-related information according to an embodiment of the present invention.

FIG. 10 is a view illustrating another example of signaling brick-related information according to an embodiment of the present invention.

FIG. 11 is a view for explaining another example of brick-related information according to an embodiment of the present invention.

FIG. 12 is a view illustrating another example of signaling brick-related information according to an embodiment of the present invention.

FIG. 13 and FIG. 14 are views for explaining another example of brick-related information according to an embodiment of the present invention.

FIG. 15 is a view illustrating another example of signaling brick-related information according to an embodiment of the present invention.

FIG. 16 and FIG. 17 are views for explaining another example of brick-related information according to an embodiment of the present invention.

FIG. 18 is a view illustrating an example of signaling for distinguishing a rectangular slice within a picture according to an embodiment of the present invention.

FIG. 19 is a view illustrating another example of signaling for distinguishing a rectangular slice within a picture according to an embodiment of the present invention.

FIG. 20 is a view illustrating another example of signaling for distinguishing a rectangular slice within a picture according to an embodiment of the present invention.

FIG. 21 is a view for explaining an image decoding method according to an embodiment of the present invention.

FIG. 22 is a view for explaining an image encoding method according to an embodiment of the present invention.

#### MODE FOR INVENTION

A variety of modifications may be made to the present invention and there are various embodiments of the present invention, examples of which will now be provided with reference to drawings and described in detail. However, the present invention is not limited thereto, although the exemplary embodiments can be construed as including all modifications, equivalents, or substitutes in a technical concept and a technical scope of the present invention. The similar reference numerals refer to the same or similar functions in various aspects. In the drawings, the shapes and dimensions of elements may be exaggerated for clarity. In the following detailed description of the present invention, references are made to the accompanying drawings that show, by way of illustration, specific embodiments in which the invention may be practiced. These embodiments are described in sufficient detail to enable those skilled in the art to implement the present disclosure. It should be understood that various embodiments of the present disclosure, although different, are not necessarily mutually exclusive. For example, specific features, structures, and characteristics described herein, in connection with one embodiment, may be implemented within other embodiments without departing from the spirit and scope of the present disclosure. In addition, it should be understood that the location or arrangement of individual elements within each disclosed embodiment may be modified without departing from the spirit and scope of the present disclosure. The following detailed description is, therefore, not to be taken in a limiting sense, and the scope of the present disclosure is defined only by the appended claims, appropriately interpreted, along with the full range of equivalents to what the claims claim.

Terms used in the specification, 'first', 'second', etc. can be used to describe various components, but the components are not to be construed as being limited to the terms. The terms are only used to differentiate one component from other components. For example, the 'first' component may be named the 'second' component without departing from the scope of the present invention, and the 'second' component may also be similarly named the 'first' component. The term 'and/or' includes a combination of a plurality of items or any one of a plurality of terms.

It will be understood that when an element is simply referred to as being 'connected to' or 'coupled to' another element without being 'directly connected to' or 'directly coupled to' another element in the present description, it may be 'directly connected to' or 'directly coupled to' another element or be connected to or coupled to another

element, having the other element intervening therebetween. In contrast, it should be understood that when an element is referred to as being “directly coupled” or “directly connected” to another element, there are no intervening elements present.

Furthermore, constitutional parts shown in the embodiments of the present invention are independently shown so as to represent characteristic functions different from each other. Thus, it does not mean that each constitutional part is constituted in a constitutional unit of separated hardware or software. In other words, each constitutional part includes each of enumerated constitutional parts for convenience. Thus, at least two constitutional parts of each constitutional part may be combined to form one constitutional part or one constitutional part may be divided into a plurality of constitutional parts to perform each function. The embodiment where each constitutional part is combined and the embodiment where one constitutional part is divided are also included in the scope of the present invention, if not departing from the essence of the present invention.

The terms used in the present specification are merely used to describe particular embodiments, and are not intended to limit the present invention. An expression used in the singular encompasses the expression of the plural, unless it has a clearly different meaning in the context. In the present specification, it is to be understood that terms such as “including”, “having”, etc. are intended to indicate the existence of the features, numbers, steps, actions, elements, parts, or combinations thereof disclosed in the specification, and are not intended to preclude the possibility that one or more other features, numbers, steps, actions, elements, parts, or combinations thereof may exist or may be added. In other words, when a specific element is referred to as being “included”, elements other than the corresponding element are not excluded, but additional elements may be included in embodiments of the present invention or the scope of the present invention.

In addition, some of constituents may not be indispensable constituents performing essential functions of the present invention but be selective constituents improving only performance thereof. The present invention may be implemented by including only the indispensable constitutional parts for implementing the essence of the present invention except the constituents used in improving performance. The structure including only the indispensable constituents except the selective constituents used in improving only performance is also included in the scope of the present invention.

Hereinafter, embodiments of the present invention will be described in detail with reference to the accompanying drawings. In describing exemplary embodiments of the present invention, well-known functions or constructions will not be described in detail since they may unnecessarily obscure the understanding of the present invention. The same constituent elements in the drawings are denoted by the same reference numerals, and a repeated description of the same elements will be omitted.

Hereinafter, an image may mean a picture configuring a video, or may mean the video itself. For example, “encoding or decoding or both of an image” may mean “encoding or decoding or both of a moving picture”, and may mean “encoding or decoding or both of one image among images of a moving picture.”

Hereinafter, terms “moving picture” and “video” may be used as the same meaning and be replaced with each other.

Hereinafter, a target image may be an encoding target image which is a target of encoding and/or a decoding target

image which is a target of decoding. Also, a target image may be an input image inputted to an encoding apparatus, and an input image inputted to a decoding apparatus. Here, a target image may have the same meaning with the current image.

Hereinafter, terms “image”, “picture”, “frame” and “screen” may be used as the same meaning and be replaced with each other.

Hereinafter, a target block may be an encoding target block which is a target of encoding and/or a decoding target block which is a target of decoding. Also, a target block may be the current block which is a target of current encoding and/or decoding. For example, terms “target block” and “current block” may be used as the same meaning and be replaced with each other.

Hereinafter, terms “block” and “unit” may be used as the same meaning and be replaced with each other. Or a “block” may represent a specific unit.

Hereinafter, terms “region” and “segment” may be replaced with each other.

Hereinafter, a specific signal may be a signal representing a specific block. For example, an original signal may be a signal representing a target block. A prediction signal may be a signal representing a prediction block. A residual signal may be a signal representing a residual block.

In embodiments, each of specific information, data, flag, index, element and attribute, etc. may have a value. A value of information, data, flag, index, element and attribute equal to “0” may represent a logical false or the first predefined value. In other words, a value “0”, a false, a logical false and the first predefined value may be replaced with each other. A value of information, data, flag, index, element and attribute equal to “1” may represent a logical true or the second predefined value. In other words, a value “1”, a true, a logical true and the second predefined value may be replaced with each other.

When a variable  $i$  or  $j$  is used for representing a column, a row or an index, a value of  $i$  may be an integer equal to or greater than 0, or equal to or greater than 1. That is, the column, the row, the index, etc. may be counted from 0 or may be counted from 1.

#### Description of Terms

Encoder: means an apparatus performing encoding. That is, means an encoding apparatus.

Decoder: means an apparatus performing decoding. That is, means a decoding apparatus.

Block: is an  $M \times N$  array of a sample. Herein,  $M$  and  $N$  may mean positive integers, and the block may mean a sample array of a two-dimensional form. The block may refer to a unit. A current block may mean an encoding target block that becomes a target when encoding, or a decoding target block that becomes a target when decoding. In addition, the current block may be at least one of an encode block, a prediction block, a residual block, and a transform block.

Sample: is a basic unit constituting a block. It may be expressed as a value from 0 to  $2^{B_d}-1$  according to a bit depth ( $B_d$ ). In the present invention, the sample may be used as a meaning of a pixel. That is, a sample, a pel, a pixel may have the same meaning with each other.

Unit: may refer to an encoding and decoding unit. When encoding and decoding an image, the unit may be a region generated by partitioning a single image. In addition, the unit may mean a subdivided unit when a single image is partitioned into subdivided units during encoding or decoding. That is, an image may be partitioned into a plurality of



units. When encoding and decoding an image, a predetermined process for each unit may be performed. A single unit may be partitioned into sub-units that have sizes smaller than the size of the unit. Depending on functions, the unit may mean a block, a macroblock, a coding tree unit, a code tree block, a coding unit, a coding block), a prediction unit, a prediction block, a residual unit), a residual block, a transform unit, a transform block, etc. In addition, in order to distinguish a unit from a block, the unit may include a luma component block, a chroma component block associated with the luma component block, and a syntax element of each color component block. The unit may have various sizes and forms, and particularly, the form of the unit may be a two-dimensional geometrical figure such as a square shape, a rectangular shape, a trapezoid shape, a triangular shape, a pentagonal shape, etc. In addition, unit information may include at least one of a unit type indicating the coding unit, the prediction unit, the transform unit, etc., and a unit size, a unit depth, a sequence of encoding and decoding of a unit, etc.

Coding Tree Unit: is configured with a single coding tree block of a luma component Y, and two coding tree blocks related to chroma components Cb and Cr. In addition, it may mean that including the blocks and a syntax element of each block. Each coding tree unit may be partitioned by using at least one of a quad-tree partitioning method, a binary-tree partitioning method and ternary-tree partitioning method to configure a lower unit such as coding unit, prediction unit, transform unit, etc. It may be used as a term for designating a sample block that becomes a process unit when encoding/decoding an image as an input image. Here, the quad-tree may mean a quaternary-tree.

When the size of the coding block is within a predetermined range, the division is possible using only quad-tree partitioning. Here, the predetermined range may be defined as at least one of a maximum size and a minimum size of a coding block in which the division is possible using only quad-tree partitioning. Information indicating a maximum/minimum size of a coding block in which quad-tree partitioning is allowed may be signaled through a bitstream, and the information may be signaled in at least one unit of a sequence, a picture parameter, a tile group, or a slice (segment). Alternatively, the maximum/minimum size of the coding block may be a fixed size predetermined in the coder/decoder. For example, when the size of the coding block corresponds to  $256 \times 256$  to  $64 \times 64$ , the division is possible only using quad-tree partitioning. Alternatively, when the size of the coding block is larger than the size of the maximum conversion block, the division is possible only using quad-tree partitioning. Herein, the block to be divided may be at least one of a coding block and a transform block. In this case, information indicating the division of the coded block (for example, split\_flag) may be a flag indicating whether or not to perform the quad-tree partitioning. When the size of the coding block falls within a predetermined range, the division is possible only using binary tree or ternary tree partitioning. In this case, the above description of the quad-tree partitioning may be applied to binary tree partitioning or ternary tree partitioning in the same manner.

Coding Tree Block: may be used as a term for designating any one of a Y coding tree block, Cb coding tree block, and Cr coding tree block.

Neighbor Block: may mean a block adjacent to a current block. The block adjacent to the current block may mean a block that comes into contact with a boundary of the current block, or a block positioned within a predetermined distance from the current block. The neighbor block may mean a

block adjacent to a vertex of the current block. Herein, the block adjacent to the vertex of the current block may mean a block vertically adjacent to a neighbor block that is horizontally adjacent to the current block, or a block horizontally adjacent to a neighbor block that is vertically adjacent to the current block.

Reconstructed Neighbor block: may mean a neighbor block adjacent to a current block and which has been already spatially/temporally encoded or decoded. Herein, the reconstructed neighbor block may mean a reconstructed neighbor unit. A reconstructed spatial neighbor block may be a block within a current picture and which has been already reconstructed through encoding or decoding or both. A reconstructed temporal neighbor block is a block at a corresponding position as the current block of the current picture within a reference image, or a neighbor block thereof.

Unit Depth: may mean a partitioned degree of a unit. In a tree structure, the highest node (Root Node) may correspond to the first unit which is not partitioned. Also, the highest node may have the least depth value. In this case, the highest node may have a depth of level 0. A node having a depth of level 1 may represent a unit generated by partitioning once the first unit. A node having a depth of level 2 may represent a unit generated by partitioning twice the first unit. A node having a depth of level n may represent a unit generated by partitioning n-times the first unit. A Leaf Node may be the lowest node and a node which cannot be partitioned further. A depth of a Leaf Node may be the maximum level. For example, a predefined value of the maximum level may be 3. A depth of a root node may be the lowest and a depth of a leaf node may be the deepest. In addition, when a unit is expressed as a tree structure, a level in which a unit is present may mean a unit depth.

Bitstream: may mean a bitstream including encoding image information.

Parameter Set: corresponds to header information among a configuration within a bitstream. At least one of a video parameter set, a sequence parameter set, a picture parameter set, and an adaptation parameter set may be included in a parameter set. In addition, a parameter set may include a slice header, a tile group header, and tile header information. The term "tile group" means a group of tiles and has the same meaning as a slice.

An adaptation parameter set may mean a parameter set that can be shared by being referred to in different pictures, subpictures, slices, tile groups, tiles, or bricks. In addition, information in an adaptation parameter set may be used by referring to different adaptation parameter sets for a subpicture, a slice, a tile group, a tile, or a brick inside a picture.

In addition, regarding the adaptation parameter set, different adaptation parameter sets may be referred to by using identifiers of different adaptation parameter sets for a subpicture, a slice, a tile group, a tile, or a brick inside a picture.

In addition, regarding the adaptation parameter set, different adaptation parameter sets may be referred to by using identifiers of different adaptation parameter sets for a slice, a tile group, a tile, or a brick inside a subpicture.

In addition, regarding the adaptation parameter set, different adaptation parameter sets may be referred to by using identifiers of different adaptation parameter sets for a tile or a brick inside a slice.

In addition, regarding the adaptation parameter set, different adaptation parameter sets may be referred to by using identifiers of different adaptation parameter sets for a brick inside a tile.

Information on an adaptation parameter set identifier may be included in a parameter set or a header of the subpicture,

and an adaptation parameter set corresponding to the adaptation parameter set identifier may be used for the subpicture.

The information on the adaptation parameter set identifier may be included in a parameter set or a header of the tile, and an adaptation parameter set corresponding to the adaptation parameter set identifier may be used for the tile.

The information on the adaptation parameter set identifier may be included in a header of the brick, and an adaptation parameter set corresponding to the adaptation parameter set identifier may be used for the brick.

The picture may be partitioned into one or more tile rows and one or more tile columns.

The subpicture may be partitioned into one or more tile rows and one or more tile columns within a picture. The subpicture may be a region having the form of a rectangle/square within a picture and may include one or more CTUs. In addition, at least one or more tiles/bricks/slices may be included within one subpicture.

The tile may be a region having the form of a rectangle/square within a picture and may include one or more CTUs. In addition, the tile may be partitioned into one or more bricks.

The brick may mean one or more CTU rows within a tile. The tile may be partitioned into one or more bricks, and each brick may have at least one or more CTU rows. A tile that is not partitioned into two or more may mean a brick.

The slice may include one or more tiles within a picture and may include one or more bricks within a tile.

Parsing: may mean determination of a value of a syntax element by performing entropy decoding, or may mean the entropy decoding itself.

Symbol: may mean at least one of a syntax element, a coding parameter, and a transform coefficient value of an encoding/decoding target unit. In addition, the symbol may mean an entropy encoding target or an entropy decoding result.

Prediction Mode: may be information indicating a mode encoded/decoded with intra prediction or a mode encoded/decoded with inter prediction.

Prediction Unit: may mean a basic unit when performing prediction such as inter-prediction, intra-prediction, inter-compensation, intra-compensation, and motion compensation. A single prediction unit may be partitioned into a plurality of partitions having a smaller size, or may be partitioned into a plurality of lower prediction units. A plurality of partitions may be a basic unit in performing prediction or compensation. A partition which is generated by dividing a prediction unit may also be a prediction unit.

Prediction Unit Partition: may mean a form obtained by partitioning a prediction unit.

Reference picture list may refer to a list including one or more reference pictures used for inter prediction or motion compensation. There are several types of usable reference picture lists, including LC (List combined), L0 (List 0), L1 (List 1), L2 (List 2), L3 (List 3).

Inter prediction indicator may refer to a direction of inter prediction (unidirectional prediction, bidirectional prediction, etc.) of a current block. Alternatively, it may refer to the number of reference pictures used to generate a prediction block of a current block. Alternatively, it may refer to the number of prediction blocks used at the time of performing inter prediction or motion compensation on a current block.

Prediction list utilization flag indicates whether a prediction block is generated using at least one reference picture in a specific reference picture list. An inter prediction indicator can be derived using a prediction list utilization flag, and

conversely, a prediction list utilization flag can be derived using an inter prediction indicator. For example, when the prediction list utilization flag has a first value of zero (0), it means that a reference picture in a reference picture list is not used to generate a prediction block. On the other hand, when the prediction list utilization flag has a second value of one (1), it means that a reference picture list is used to generate a prediction block.

Reference picture index may refer to an index indicating a specific reference picture in a reference picture list.

Reference picture may mean a reference picture which is referred to by a specific block for the purposes of inter prediction or motion compensation of the specific block. Alternatively, the reference picture may be a picture including a reference block referred to by a current block for inter prediction or motion compensation. Hereinafter, the terms "reference picture" and "reference image" have the same meaning and can be interchangeably.

Motion vector may be a two-dimensional vector used for inter prediction or motion compensation. The motion vector may mean an offset between an encoding/decoding target block and a reference block. For example, (mvX, mvY) may represent a motion vector. Here, mvX may represent a horizontal component and mvY may represent a vertical component.

Search range may be a two-dimensional region which is searched to retrieve a motion vector during inter prediction. For example, the size of the search range may be MxN. Here, M and N are both integers.

Motion vector candidate may refer to a prediction candidate block or a motion vector of the prediction candidate block when predicting a motion vector. In addition, a motion vector candidate may be included in a motion vector candidate list.

Motion vector candidate list may mean a list composed of one or more motion vector candidates.

Motion vector candidate index may mean an indicator indicating a motion vector candidate in a motion vector candidate list. Alternatively, it may be an index of a motion vector predictor.

Motion information may mean information including at least one of the items including a motion vector, a reference picture index, an inter prediction indicator, a prediction list utilization flag, reference picture list information, a reference picture, a motion vector candidate, a motion vector candidate index, a merge candidate, and a merge index.

Merge candidate list may mean a list composed of one or more merge candidates.

Merge candidate may mean a spatial merge candidate, a temporal merge candidate, a combined merge candidate, a combined bi-predictive merge candidate, or a zero merge candidate. The merge candidate may include motion information such as an inter prediction indicator, a reference picture index for each list, a motion vector, a prediction list utilization flag, and an inter prediction indicator.

Merge index may mean an indicator indicating a merge candidate in a merge candidate list. Alternatively, the merge index may indicate a block from which a merge candidate has been derived, among reconstructed blocks spatially/temporally adjacent to a current block. Alternatively, the merge index may indicate at least one piece of motion information of a merge candidate.

Transform Unit: may mean a basic unit when performing encoding/decoding such as transform, inverse-transform, quantization, dequantization, transform coefficient encoding/decoding of a residual signal. A single transform unit may be partitioned into a plurality of lower-level transform

## 11

units having a smaller size. Here, transformation/inverse-transformation may comprise at least one among the first transformation/the first inverse-transformation and the second transformation/the second inverse-transformation.

Scaling: may mean a process of multiplying a quantized level by a factor. A transform coefficient may be generated by scaling a quantized level. The scaling also may be referred to as dequantization.

Quantization Parameter: may mean a value used when generating a quantized level using a transform coefficient during quantization. The quantization parameter also may mean a value used when generating a transform coefficient by scaling a quantized level during dequantization. The quantization parameter may be a value mapped on a quantization step size.

Delta Quantization Parameter: may mean a difference value between a predicted quantization parameter and a quantization parameter of an encoding/decoding target unit.

Scan: may mean a method of sequencing coefficients within a unit, a block or a matrix. For example, changing a two-dimensional matrix of coefficients into a one-dimensional matrix may be referred to as scanning, and changing a one-dimensional matrix of coefficients into a two-dimensional matrix may be referred to as scanning or inverse scanning.

Transform Coefficient: may mean a coefficient value generated after transform is performed in an encoder. It may mean a coefficient value generated after at least one of entropy decoding and dequantization is performed in a decoder. A quantized level obtained by quantizing a transform coefficient or a residual signal, or a quantized transform coefficient level also may fall within the meaning of the transform coefficient.

Quantized Level: may mean a value generated by quantizing a transform coefficient or a residual signal in an encoder. Alternatively, the quantized level may mean a value that is a dequantization target to undergo dequantization in a decoder. Similarly, a quantized transform coefficient level that is a result of transform and quantization also may fall within the meaning of the quantized level.

Non-zero Transform Coefficient: may mean a transform coefficient having a value other than zero, or a transform coefficient level or a quantized level having a value other than zero.

Quantization Matrix: may mean a matrix used in a quantization process or a dequantization process performed to improve subjective or objective image quality. The quantization matrix also may be referred to as a scaling list.

Quantization Matrix Coefficient: may mean each element within a quantization matrix. The quantization matrix coefficient also may be referred to as a matrix coefficient.

Default Matrix: may mean a predetermined quantization matrix preliminarily defined in an encoder or a decoder.

Non-default Matrix: may mean a quantization matrix that is not preliminarily defined in an encoder or a decoder but is signaled by a user.

Statistic Value: a statistic value for at least one among a variable, a coding parameter, a constant value, etc. which have a computable specific value may be one or more among an average value, a sum value, a weighted average value, a weighted sum value, the minimum value, the maximum value, the most frequent value, a median value, an interpolated value of the corresponding specific values.

FIG. 1 is a block diagram showing a configuration of an encoding apparatus according to an embodiment to which the present invention is applied.

## 12

An encoding apparatus 100 may be an encoder, a video encoding apparatus, or an image encoding apparatus. A video may include at least one image. The encoding apparatus 100 may sequentially encode at least one image.

Referring to FIG. 1, the encoding apparatus 100 may include a motion prediction unit 111, a motion compensation unit 112, an intra-prediction unit 120, a switch 115, a subtractor 125, a transform unit 130, a quantization unit 140, an entropy encoding unit 150, a dequantization unit 160, an inverse-transform unit 170, an adder 175, a filter unit 180, and a reference picture buffer 190.

The encoding apparatus 100 may perform encoding of an input image by using an intra mode or an inter mode or both. In addition, encoding apparatus 100 may generate a bitstream including encoded information through encoding the input image, and output the generated bitstream. The generated bitstream may be stored in a computer readable recording medium, or may be streamed through a wired/wireless transmission medium. When an intra mode is used as a prediction mode, the switch 115 may be switched to an intra. Alternatively, when an inter mode is used as a prediction mode, the switch 115 may be switched to an inter mode. Herein, the intra mode may mean an intra-prediction mode, and the inter mode may mean an inter-prediction mode. The encoding apparatus 100 may generate a prediction block for an input block of the input image. In addition, the encoding apparatus 100 may encode a residual block using a residual of the input block and the prediction block after the prediction block being generated. The input image may be called as a current image that is a current encoding target. The input block may be called as a current block that is current encoding target, or as an encoding target block.

When a prediction mode is an intra mode, the intra-prediction unit 120 may use a sample of a block that has been already encoded/decoded and is adjacent to a current block as a reference sample. The intra-prediction unit 120 may perform spatial prediction for the current block by using a reference sample, or generate prediction samples of an input block by performing spatial prediction. Herein, the intra prediction may mean intra-prediction.

When a prediction mode is an inter mode, the motion prediction unit 111 may retrieve a region that best matches with an input block from a reference image when performing motion prediction, and deduce a motion vector by using the retrieved region. In this case, a search region may be used as the region. The reference image may be stored in the reference picture buffer 190. Here, when encoding/decoding for the reference image is performed, it may be stored in the reference picture buffer 190.

The motion compensation unit 112 may generate a prediction block by performing motion compensation for the current block using a motion vector. Herein, inter-prediction may mean inter-prediction or motion compensation.

When the value of the motion vector is not an integer, the motion prediction unit 111 and the motion compensation unit 112 may generate the prediction block by applying an interpolation filter to a partial region of the reference picture. In order to perform inter-picture prediction or motion compensation on a coding unit, it may be determined that which mode among a skip mode, a merge mode, an advanced motion vector prediction (AMVP) mode, and a current picture referring mode is used for motion prediction and motion compensation of a prediction unit included in the corresponding coding unit. Then, inter-picture prediction or motion compensation may be differently performed depending on the determined mode.

## 13

The subtractor **125** may generate a residual block by using a difference of an input block and a prediction block. The residual block may be called as a residual signal. The residual signal may mean a difference between an original signal and a prediction signal. In addition, the residual signal may be a signal generated by transforming or quantizing, or transforming and quantizing a difference between the original signal and the prediction signal. The residual block may be a residual signal of a block unit.

The transform unit **130** may generate a transform coefficient by performing transform of a residual block, and output the generated transform coefficient. Herein, the transform coefficient may be a coefficient value generated by performing transform of the residual block. When a transform skip mode is applied, the transform unit **130** may skip transform of the residual block.

A quantized level may be generated by applying quantization to the transform coefficient or to the residual signal. Hereinafter, the quantized level may be also called as a transform coefficient in embodiments.

The quantization unit **140** may generate a quantized level by quantizing the transform coefficient or the residual signal according to a parameter, and output the generated quantized level. Herein, the quantization unit **140** may quantize the transform coefficient by using a quantization matrix.

The entropy encoding unit **150** may generate a bitstream by performing entropy encoding according to a probability distribution on values calculated by the quantization unit **140** or on coding parameter values calculated when performing encoding, and output the generated bitstream. The entropy encoding unit **150** may perform entropy encoding of sample information of an image and information for decoding an image. For example, the information for decoding the image may include a syntax element.

When entropy encoding is applied, symbols are represented so that a smaller number of bits are assigned to a symbol having a high chance of being generated and a larger number of bits are assigned to a symbol having a low chance of being generated, and thus, the size of bit stream for symbols to be encoded may be decreased. The entropy encoding unit **150** may use an encoding method for entropy encoding such as exponential Golomb, context-adaptive variable length coding (CAVLC), context-adaptive binary arithmetic coding (CABAC), etc. For example, the entropy encoding unit **150** may perform entropy encoding by using a variable length coding/code (VLC) table. In addition, the entropy encoding unit **150** may deduce a binarization method of a target symbol and a probability model of a target symbol/bin, and perform arithmetic coding by using the deduced binarization method, and a context model.

In order to encode a transform coefficient level (quantized level), the entropy encoding unit **150** may change a two-dimensional block form coefficient into a one-dimensional vector form by using a transform coefficient scanning method.

A coding parameter may include information (flag, index, etc.) such as syntax element that is encoded in an encoder and signaled to a decoder, and information derived when performing encoding or decoding. The coding parameter may mean information required when encoding or decoding an image. For example, at least one value or a combination form of a unit/block size, a unit/block depth, unit/block partition information, unit/block shape, unit/block partition structure, whether to partition of a quad-tree form, whether to partition of a binary-tree form, a partition direction of a binary-tree form (horizontal direction or vertical direction), a partition form of a binary-tree form (symmetric partition or

## 14

asymmetric partition), whether or not a current coding unit is partitioned by ternary tree partitioning, direction (horizontal or vertical direction) of the ternary tree partitioning, type (symmetric or asymmetric type) of the ternary tree partitioning, whether a current coding unit is partitioned by multi-type tree partitioning, direction (horizontal or vertical direction) of the multi-type tree partitioning, type (symmetric or asymmetric type) of the multi-type tree partitioning, and a tree (binary tree or ternary tree) structure of the multi-type tree partitioning, a prediction mode (intra prediction or inter prediction), a luma intra-prediction mode/direction, a chroma intra-prediction mode/direction, intra partition information, inter partition information, a coding block partition flag, a prediction block partition flag, a transform block partition flag, a reference sample filtering method, a reference sample filter tab, a reference sample filter coefficient, a prediction block filtering method, a prediction block filter tap, a prediction block filter coefficient, a prediction block boundary filtering method, a prediction block boundary filter tab, a prediction block boundary filter coefficient, an intra-prediction mode, an inter-prediction mode, motion information, a motion vector, a motion vector difference, a reference picture index, a inter-prediction angle, an inter-prediction indicator, a prediction list utilization flag, a reference picture list, a reference picture, a motion vector predictor index, a motion vector predictor candidate, a motion vector candidate list, whether to use a merge mode, a merge index, a merge candidate, a merge candidate list, whether to use a skip mode, an interpolation filter type, an interpolation filter tab, an interpolation filter coefficient, a motion vector size, a presentation accuracy of a motion vector, a transform type, a transform size, information of whether or not a primary (first) transform is used, information of whether or not a secondary transform is used, a primary transform index, a secondary transform index, information of whether or not a residual signal is present, a coded block pattern, a coded block flag (CBF), a quantization parameter, a quantization parameter residue, a quantization matrix, whether to apply an intra loop filter, an intra loop filter coefficient, an intra loop filter tab, an intra loop filter shape/form, whether to apply a deblocking filter, a deblocking filter coefficient, a deblocking filter tab, a deblocking filter strength, a deblocking filter shape/form, whether to apply an adaptive sample offset, an adaptive sample offset value, an adaptive sample offset category, an adaptive sample offset type, whether to apply an adaptive loop filter, an adaptive loop filter coefficient, an adaptive loop filter tab, an adaptive loop filter shape/form, a binarization/inverse-binarization method, a context model determining method, a context model updating method, whether to perform a regular mode, whether to perform a bypass mode, a context bin, a bypass bin, a significant coefficient flag, a last significant coefficient flag, a coded flag for a unit of a coefficient group, a position of the last significant coefficient, a flag for whether a value of a coefficient is larger than 1, a flag for whether a value of a coefficient is larger than 2, a flag for whether a value of a coefficient is larger than 3, information on a remaining coefficient value, a sign information, a reconstructed luma sample, a reconstructed chroma sample, a residual luma sample, a residual chroma sample, a luma transform coefficient, a chroma transform coefficient, a quantized luma level, a quantized chroma level, a transform coefficient level scanning method, a size of a motion vector search area at a decoder side, a shape of a motion vector search area at a decoder side, a number of time of a motion vector search at a decoder side, information on a CTU size, information on

a minimum block size, information on a maximum block size, information on a maximum block depth, information on a minimum block depth, an image displaying/outputting sequence, slice identification information, a slice type, slice partition information, tile identification information, a tile type, tile partition information, tile group identification information, a tile group type, tile group partition information, a picture type, a bit depth of an input sample, a bit depth of a reconstruction sample, a bit depth of a residual sample, a bit depth of a transform coefficient, a bit depth of a quantized level, and information on a luma signal or information on a chroma signal may be included in the coding parameter.

Herein, signaling the flag or index may mean that a corresponding flag or index is entropy encoded and included in a bitstream by an encoder, and may mean that the corresponding flag or index is entropy decoded from a bitstream by a decoder.

When the encoding apparatus 100 performs encoding through inter-prediction, an encoded current image may be used as a reference image for another image that is processed afterwards. Accordingly, the encoding apparatus 100 may reconstruct or decode the encoded current image, or store the reconstructed or decoded image as a reference image in reference picture buffer 190.

A quantized level may be dequantized in the dequantization unit 160, or may be inverse-transformed in the inverse-transform unit 170. A dequantized or inverse-transformed coefficient or both may be added with a prediction block by the adder 175. By adding the dequantized or inverse-transformed coefficient or both with the prediction block, a reconstructed block may be generated. Herein, the dequantized or inverse-transformed coefficient or both may mean a coefficient on which at least one of dequantization and inverse-transform is performed, and may mean a reconstructed residual block.

A reconstructed block may pass through the filter unit 180. The filter unit 180 may apply at least one of a deblocking filter, a sample adaptive offset (SAO), and an adaptive loop filter (ALF) to a reconstructed sample, a reconstructed block or a reconstructed image. The filter unit 180 may be called as an in-loop filter.

The deblocking filter may remove block distortion generated in boundaries between blocks. In order to determine whether or not to apply a deblocking filter, whether or not to apply a deblocking filter to a current block may be determined based samples included in several rows or columns which are included in the block. When a deblocking filter is applied to a block, another filter may be applied according to a required deblocking filtering strength.

In order to compensate an encoding error, a proper offset value may be added to a sample value by using a sample adaptive offset. The sample adaptive offset may correct an offset of a deblocked image from an original image by a sample unit. A method of partitioning samples of an image into a predetermined number of regions, determining a region to which an offset is applied, and applying the offset to the determined region, or a method of applying an offset in consideration of edge information on each sample may be used.

The adaptive loop filter may perform filtering based on a comparison result of the filtered reconstructed image and the original image. Samples included in an image may be partitioned into predetermined groups, a filter to be applied to each group may be determined, and differential filtering may be performed for each group. Information of whether or

not to apply the ALF may be signaled by coding units (CUs), and a form and coefficient of the ALF to be applied to each block may vary.

The reconstructed block or the reconstructed image having passed through the filter unit 180 may be stored in the reference picture buffer 190. A reconstructed block processed by the filter unit 180 may be a part of a reference image. That is, a reference image is a reconstructed image composed of reconstructed blocks processed by the filter unit 180. The stored reference image may be used later in inter prediction or motion compensation.

FIG. 2 is a block diagram showing a configuration of a decoding apparatus according to an embodiment and to which the present invention is applied.

A decoding apparatus 200 may be a decoder, a video decoding apparatus, or an image decoding apparatus.

Referring to FIG. 2, the decoding apparatus 200 may include an entropy decoding unit 210, a dequantization unit 220, an inverse-transform unit 230, an intra-prediction unit 240, a motion compensation unit 250, an adder 225, a filter unit 260, and a reference picture buffer 270.

The decoding apparatus 200 may receive a bitstream output from the encoding apparatus 100. The decoding apparatus 200 may receive a bitstream stored in a computer readable recording medium, or may receive a bitstream that is streamed through a wired/wireless transmission medium. The decoding apparatus 200 may decode the bitstream by using an intra mode or an inter mode. In addition, the decoding apparatus 200 may generate a reconstructed image generated through decoding or a decoded image, and output the reconstructed image or decoded image.

When a prediction mode used when decoding is an intra mode, a switch may be switched to an intra. Alternatively, when a prediction mode used when decoding is an inter mode, a switch may be switched to an inter mode.

The decoding apparatus 200 may obtain a reconstructed residual block by decoding the input bitstream, and generate a prediction block. When the reconstructed residual block and the prediction block are obtained, the decoding apparatus 200 may generate a reconstructed block that becomes a decoding target by adding the reconstructed residual block with the prediction block. The decoding target block may be called a current block.

The entropy decoding unit 210 may generate symbols by entropy decoding the bitstream according to a probability distribution. The generated symbols may include a symbol of a quantized level form. Herein, an entropy decoding method may be an inverse-process of the entropy encoding method described above.

In order to decode a transform coefficient level (quantized level), the entropy decoding unit 210 may change a one-dimensional vector form coefficient into a two-dimensional block form by using a transform coefficient scanning method.

A quantized level may be dequantized in the dequantization unit 220, or inverse-transformed in the inverse-transform unit 230. The quantized level may be a result of dequantizing or inverse-transforming or both, and may be generated as a reconstructed residual block. Herein, the dequantization unit 220 may apply a quantization matrix to the quantized level.

When an intra mode is used, the intra-prediction unit 240 may generate a prediction block by performing, for the current block, spatial prediction that uses a sample value of a block adjacent to a decoding target block and which has been already decoded.

When an inter mode is used, the motion compensation unit **250** may generate a prediction block by performing, for the current block, motion compensation that uses a motion vector and a reference image stored in the reference picture buffer **270**.

The adder **225** may generate a reconstructed block by adding the reconstructed residual block with the prediction block. The filter unit **260** may apply at least one of a deblocking filter, a sample adaptive offset, and an adaptive loop filter to the reconstructed block or reconstructed image. The filter unit **260** may output the reconstructed image. The reconstructed block or reconstructed image may be stored in the reference picture buffer **270** and used when performing inter-prediction. A reconstructed block processed by the filter unit **260** may be a part of a reference image. That is, a reference image is a reconstructed image composed of reconstructed blocks processed by the filter unit **260**. The stored reference image may be used later in inter prediction or motion compensation.

FIG. 3 is a view schematically showing a partition structure of an image when encoding and decoding the image. FIG. 3 schematically shows an example of partitioning a single unit into a plurality of lower units.

In order to efficiently partition an image, when encoding and decoding, a coding unit (CU) may be used. The coding unit may be used as a basic unit when encoding/decoding the image. In addition, the coding unit may be used as a unit for distinguishing an intra prediction mode and an inter prediction mode when encoding/decoding the image. The coding unit may be a basic unit used for prediction, transform, quantization, inverse-transform, dequantization, or an encoding/decoding process of a transform coefficient.

Referring to FIG. 3, an image **300** is sequentially partitioned in a largest coding unit (LCU), and a LCU unit is determined as a partition structure. Herein, the LCU may be used in the same meaning as a coding tree unit (CTU). A unit partitioning may mean partitioning a block associated with to the unit. In block partition information, information of a unit depth may be included. Depth information may represent a number of times or a degree or both in which a unit is partitioned. A single unit may be partitioned into a plurality of lower level units hierarchically associated with depth information based on a tree structure. In other words, a unit and a lower level unit generated by partitioning the unit may correspond to a node and a child node of the node, respectively. Each of partitioned lower unit may have depth information. Depth information may be information representing a size of a CU, and may be stored in each CU. Unit depth represents times and/or degrees related to partitioning a unit. Therefore, partitioning information of a lower-level unit may comprise information on a size of the lower-level unit.

A partition structure may mean a distribution of a coding unit (CU) within an LCU **310**. Such a distribution may be determined according to whether or not to partition a single CU into a plurality (positive integer equal to or greater than 2 including 2, 4, 8, 16, etc.) of CUs. A horizontal size and a vertical size of the CU generated by partitioning may respectively be half of a horizontal size and a vertical size of the CU before partitioning, or may respectively have sizes smaller than a horizontal size and a vertical size before partitioning according to a number of times of partitioning. The CU may be recursively partitioned into a plurality of CUs. By the recursive partitioning, at least one among a height and a width of a CU after partitioning may decrease comparing with at least one among a height and a width of a CU before partitioning. Partitioning of the CU may be

recursively performed until to a predefined depth or predefined size. For example, a depth of an LCU may be 0, and a depth of a smallest coding unit (SCU) may be a predefined maximum depth. Herein, the LCU may be a coding unit having a maximum coding unit size, and the SCU may be a coding unit having a minimum coding unit size as described above. Partitioning is started from the LCU **310**, a CU depth increases by 1 as a horizontal size or a vertical size or both of the CU decreases by partitioning. For example, for each depth, a CU which is not partitioned may have a size of  $2N \times 2N$ . Also, in case of a CU which is partitioned, a CU with a size of  $2N \times 2N$  may be partitioned into four CUs with a size of  $N \times N$ . A size of  $N$  may decrease to half as a depth increase by 1.

In addition, information whether or not the CU is partitioned may be represented by using partition information of the CU. The partition information may be 1-bit information. All CUs, except for a SCU, may include partition information. For example, when a value of partition information is a first value, the CU may not be partitioned, when a value of partition information is a second value, the CU may be partitioned.

Referring to FIG. 3, an LCU having a depth 0 may be a  $64 \times 64$  block. 0 may be a minimum depth. A SCU having a depth 3 may be an  $8 \times 8$  block. 3 may be a maximum depth. A CU of a  $32 \times 32$  block and a  $16 \times 16$  block may be respectively represented as a depth 1 and a depth 2.

For example, when a single coding unit is partitioned into four coding units, a horizontal size and a vertical size of the four partitioned coding units may be a half size of a horizontal and vertical size of the CU before being partitioned. In one embodiment, when a coding unit having a  $32 \times 32$  size is partitioned into four coding units, each of the four partitioned coding units may have a  $16 \times 16$  size. When a single coding unit is partitioned into four coding units, it may be called that the coding unit may be partitioned into a quad-tree form.

For example, when one coding unit is partitioned into two sub-coding units, the horizontal or vertical size (width or height) of each of the two sub-coding units may be half the horizontal or vertical size of the original coding unit. For example, when a coding unit having a size of  $32 \times 32$  is vertically partitioned into two sub-coding units, each of the two sub-coding units may have a size of  $16 \times 32$ . For example, when a coding unit having a size of  $8 \times 32$  is horizontally partitioned into two sub-coding units, each of the two sub-coding units may have a size of  $8 \times 16$ . When one coding unit is partitioned into two sub-coding units, it can be said that the coding unit is binary-partitioned or is partitioned by a binary tree partition structure.

For example, when one coding unit is partitioned into three sub-coding units, the horizontal or vertical size of the coding unit can be partitioned with a ratio of 1:2:1, thereby producing three sub-coding units whose horizontal or vertical sizes are in a ratio of 1:2:1. For example, when a coding unit having a size of  $16 \times 32$  is horizontally partitioned into three sub-coding units, the three sub-coding units may have sizes of  $16 \times 8$ ,  $16 \times 16$ , and  $16 \times 8$  respectively, in the order from the uppermost to the lowermost sub-coding unit. For example, when a coding unit having a size of  $32 \times 32$  is vertically split into three sub-coding units, the three sub-coding units may have sizes of  $8 \times 32$ ,  $16 \times 32$ , and  $8 \times 32$ , respectively in the order from the left to the right sub-coding unit. When one coding unit is partitioned into three sub-coding units, it can be said that the coding unit is ternary-partitioned or partitioned by a ternary tree partition structure.

In FIG. 3, a coding tree unit (CTU) 320 is an example of a CTU to which a quad tree partition structure, a binary tree partition structure, and a ternary tree partition structure are all applied.

As described above, in order to partition the CTU, at least one of a quad tree partition structure, a binary tree partition structure, and a ternary tree partition structure may be applied. Various tree partition structures may be sequentially applied to the CTU, according to a predetermined priority order. For example, the quad tree partition structure may be preferentially applied to the CTU. A coding unit that cannot be partitioned any longer using a quad tree partition structure may correspond to a leaf node of a quad tree. A coding unit corresponding to a leaf node of a quad tree may serve as a root node of a binary and/or ternary tree partition structure. That is, a coding unit corresponding to a leaf node of a quad tree may be further partitioned by a binary tree partition structure or a ternary tree partition structure, or may not be further partitioned. Therefore, by preventing a coding block that results from binary tree partitioning or ternary tree partitioning of a coding unit corresponding to a leaf node of a quad tree from undergoing further quad tree partitioning, block partitioning and/or signaling of partition information can be effectively performed.

The fact that a coding unit corresponding to a node of a quad tree is partitioned may be signaled using quad partition information. The quad partition information having a first value (e.g., "1") may indicate that a current coding unit is partitioned by the quad tree partition structure. The quad partition information having a second value (e.g., "0") may indicate that a current coding unit is not partitioned by the quad tree partition structure. The quad partition information may be a flag having a predetermined length (e.g., one bit).

There may not be a priority between the binary tree partitioning and the ternary tree partitioning. That is, a coding unit corresponding to a leaf node of a quad tree may further undergo arbitrary partitioning among the binary tree partitioning and the ternary tree partitioning. In addition, a coding unit generated through the binary tree partitioning or the ternary tree partitioning may undergo a further binary tree partitioning or a further ternary tree partitioning, or may not be further partitioned.

A tree structure in which there is no priority among the binary tree partitioning and the ternary tree partitioning is referred to as a multi-type tree structure. A coding unit corresponding to a leaf node of a quad tree may serve as a root node of a multi-type tree. Whether to partition a coding unit which corresponds to a node of a multi-type tree may be signaled using at least one of multi-type tree partition indication information, partition direction information, and partition tree information. For partitioning of a coding unit corresponding to a node of a multi-type tree, the multi-type tree partition indication information, the partition direction, and the partition tree information may be sequentially signaled.

The multi-type tree partition indication information having a first value (e.g., "1") may indicate that a current coding unit is to undergo a multi-type tree partitioning. The multi-type tree partition indication information having a second value (e.g., "0") may indicate that a current coding unit is not to undergo a multi-type tree partitioning.

When a coding unit corresponding to a node of a multi-type tree is further partitioned by a multi-type tree partition structure, the coding unit may include partition direction information. The partition direction information may indicate in which direction a current coding unit is to be partitioned for the multi-type tree partitioning. The partition

direction information having a first value (e.g., "1") may indicate that a current coding unit is to be vertically partitioned. The partition direction information having a second value (e.g., "0") may indicate that a current coding unit is to be horizontally partitioned.

When a coding unit corresponding to a node of a multi-type tree is further partitioned by a multi-type tree partition structure, the current coding unit may include partition tree information. The partition tree information may indicate a tree partition structure which is to be used for partitioning of a node of a multi-type tree. The partition tree information having a first value (e.g., "1") may indicate that a current coding unit is to be partitioned by a binary tree partition structure. The partition tree information having a second value (e.g., "0") may indicate that a current coding unit is to be partitioned by a ternary tree partition structure.

The partition indication information, the partition tree information, and the partition direction information may each be a flag having a predetermined length (e.g., one bit).

At least any one of the quadtree partition indication information, the multi-type tree partition indication information, the partition direction information, and the partition tree information may be entropy encoded/decoded. For the entropy-encoding/decoding of those types of information, information on a neighboring coding unit adjacent to the current coding unit may be used. For example, there is a high probability that the partition type (the partitioned or non-partitioned, the partition tree, and/or the partition direction) of a left neighboring coding unit and/or an upper neighboring coding unit of a current coding unit is similar to that of the current coding unit. Therefore, context information for entropy encoding/decoding of the information on the current coding unit may be derived from the information on the neighboring coding units. The information on the neighboring coding units may include at least any one of quad partition information, multi-type tree partition indication information, partition direction information, and partition tree information.

As another example, among binary tree partitioning and ternary tree partitioning, binary tree partitioning may be preferentially performed. That is, a current coding unit may primarily undergo binary tree partitioning, and then a coding unit corresponding to a leaf node of a binary tree may be set as a root node for ternary tree partitioning. In this case, neither quad tree partitioning nor binary tree partitioning may not be performed on the coding unit corresponding to a node of a ternary tree.

A coding unit that cannot be partitioned by a quad tree partition structure, a binary tree partition structure, and/or a ternary tree partition structure becomes a basic unit for coding, prediction and/or transformation. That is, the coding unit cannot be further partitioned for prediction and/or transformation. Therefore, the partition structure information and the partition information used for partitioning a coding unit into prediction units and/or transformation units may not be present in a bit stream.

However, when the size of a coding unit (i.e., a basic unit for partitioning) is larger than the size of a maximum transformation block, the coding unit may be recursively partitioned until the size of the coding unit is reduced to be equal to or smaller than the size of the maximum transformation block. For example, when the size of a coding unit is 64×64 and when the size of a maximum transformation block is 32×32, the coding unit may be partitioned into four 32×32 blocks for transformation. For example, when the size of a coding unit is 32×64 and the size of a maximum transformation block is 32×32, the coding unit may be

partitioned into two 32×32 blocks for the transformation. In this case, the partitioning of the coding unit for transformation is not signaled separately, and may be determined through comparison between the horizontal or vertical size of the coding unit and the horizontal or vertical size of the maximum transformation block. For example, when the horizontal size (width) of the coding unit is larger than the horizontal size (width) of the maximum transformation block, the coding unit may be vertically bisected. For example, when the vertical size (height) of the coding unit is larger than the vertical size (height) of the maximum transformation block, the coding unit may be horizontally bisected.

Information of the maximum and/or minimum size of the coding unit and information of the maximum and/or minimum size of the transformation block may be signaled or determined at an upper level of the coding unit. The upper level may be, for example, a sequence level, a picture level, a slice level, a tile group level, a tile level, or the like. For example, the minimum size of the coding unit may be determined to be 4×4. For example, the maximum size of the transformation block may be determined to be 64×64. For example, the minimum size of the transformation block may be determined to be 4×4.

Information of the minimum size (quad tree minimum size) of a coding unit corresponding to a leaf node of a quad tree and/or information of the maximum depth (the maximum tree depth of a multi-type tree) from a root node to a leaf node of the multi-type tree may be signaled or determined at an upper level of the coding unit. For example, the upper level may be a sequence level, a picture level, a slice level, a tile group level, a tile level, or the like. Information of the minimum size of a quad tree and/or information of the maximum depth of a multi-type tree may be signaled or determined for each of an intra-picture slice and an inter-picture slice.

Difference information between the size of a CTU and the maximum size of a transformation block may be signaled or determined at an upper level of the coding unit. For example, the upper level may be a sequence level, a picture level, a slice level, a tile group level, a tile level, or the like. Information of the maximum size of the coding units corresponding to the respective nodes of a binary tree (hereinafter, referred to as a maximum size of a binary tree) may be determined based on the size of the coding tree unit and the difference information. The maximum size of the coding units corresponding to the respective nodes of a ternary tree (hereinafter, referred to as a maximum size of a ternary tree) may vary depending on the type of slice. For example, for an intra-picture slice, the maximum size of a ternary tree may be 32×32. For example, for an inter-picture slice, the maximum size of a ternary tree may be 128×128. For example, the minimum size of the coding units corresponding to the respective nodes of a binary tree (hereinafter, referred to as a minimum size of a binary tree) and/or the minimum size of the coding units corresponding to the respective nodes of a ternary tree (hereinafter, referred to as a minimum size of a ternary tree) may be set as the minimum size of a coding block.

As another example, the maximum size of a binary tree and/or the maximum size of a ternary tree may be signaled or determined at the slice level. Alternatively, the minimum size of the binary tree and/or the minimum size of the ternary tree may be signaled or determined at the slice level.

Depending on size and depth information of the above-described various blocks, quad partition information, multi-type tree partition indication information, partition tree

information and/or partition direction information may be included or may not be included in a bit stream.

For example, when the size of the coding unit is not larger than the minimum size of a quad tree, the coding unit does not contain quad partition information. Thus, the quad partition information may be deduced from a second value.

For example, when the sizes (horizontal and vertical sizes) of a coding unit corresponding to a node of a multi-type tree are larger than the maximum sizes (horizontal and vertical sizes) of a binary tree and/or the maximum sizes (horizontal and vertical sizes) of a ternary tree, the coding unit may not be binary-partitioned or ternary-partitioned. Accordingly, the multi-type tree partition indication information may not be signaled but may be deduced from a second value.

Alternatively, when the sizes (horizontal and vertical sizes) of a coding unit corresponding to a node of a multi-type tree are the same as the maximum sizes (horizontal and vertical sizes) of a binary tree and/or are two times as large as the maximum sizes (horizontal and vertical sizes) of a ternary tree, the coding unit may not be further binary-partitioned or ternary-partitioned. Accordingly, the multi-type tree partition indication information may not be signaled but be derived from a second value. This is because when a coding unit is partitioned by a binary tree partition structure and/or a ternary tree partition structure, a coding unit smaller than the minimum size of a binary tree and/or the minimum size of a ternary tree is generated.

Alternatively, the binary tree partitioning or the ternary tree partitioning may be limited on the basis of the size of a virtual pipeline data unit (hereinafter, a pipeline buffer size). For example, when the coding unit is divided into sub-coding units which do not fit the pipeline buffer size by the binary tree partitioning or the ternary tree partitioning, the corresponding binary tree partitioning or ternary tree partitioning may be limited. The pipeline buffer size may be the size of the maximum transform block (e.g., 64×64). For example, when the pipeline buffer size is 64×64, the division below may be limited.

N×M (N and/or M is 128) Ternary tree partitioning for coding units

128×N (N≤64) Binary tree partitioning in horizontal direction for coding units

N×128 (N≤64) Binary tree partitioning in vertical direction for coding units

Alternatively, when the depth of a coding unit corresponding to a node of a multi-type tree is equal to the maximum depth of the multi-type tree, the coding unit may not be further binary-partitioned and/or ternary-partitioned. Accordingly, the multi-type tree partition indication information may not be signaled but may be deduced from a second value.

Alternatively, only when at least one of vertical direction binary tree partitioning, horizontal direction binary tree partitioning, vertical direction ternary tree partitioning, and horizontal direction ternary tree partitioning is possible for a coding unit corresponding to a node of a multi-type tree, the multi-type tree partition indication information may be signaled. Otherwise, the coding unit may not be binary-partitioned and/or ternary-partitioned. Accordingly, the multi-type tree partition indication information may not be signaled but may be deduced from a second value.

Alternatively, only when both of the vertical direction binary tree partitioning and the horizontal direction binary tree partitioning or both of the vertical direction ternary tree partitioning and the horizontal direction ternary tree partitioning are possible for a coding unit corresponding to a



node of a multi-type tree, the partition direction information may be signaled. Otherwise, the partition direction information may not be signaled but may be derived from a value indicating possible partitioning directions.

Alternatively, only when both of the vertical direction binary tree partitioning and the vertical direction ternary tree partitioning or both of the horizontal direction binary tree partitioning and the horizontal direction ternary tree partitioning are possible for a coding tree corresponding to a node of a multi-type tree, the partition tree information may be signaled. Otherwise, the partition tree information may not be signaled but be deduced from a value indicating a possible partitioning tree structure.

FIG. 4 is a view showing an intra-prediction process.

Arrows from center to outside in FIG. 4 may represent prediction directions of intra prediction modes.

Intra encoding and/or decoding may be performed by using a reference sample of a neighbor block of the current block. A neighbor block may be a reconstructed neighbor block. For example, intra encoding and/or decoding may be performed by using a coding parameter or a value of a reference sample included in a reconstructed neighbor block.

A prediction block may mean a block generated by performing intra prediction. A prediction block may correspond to at least one among CU, PU and TU. A unit of a prediction block may have a size of one among CU, PU and TU. A prediction block may be a square block having a size of 2×2, 4×4, 16×16, 32×32 or 64×64 etc. or may be a rectangular block having a size of 2×8, 4×8, 2×16, 4×16 and 8×16 etc.

Intra prediction may be performed according to intra prediction mode for the current block. The number of intra prediction modes which the current block may have may be a fixed value and may be a value determined differently according to an attribute of a prediction block. For example, an attribute of a prediction block may comprise a size of a prediction block and a shape of a prediction block, etc.

The number of intra-prediction modes may be fixed to N regardless of a block size. Or, the number of intra prediction modes may be 3, 5, 9, 17, 34, 35, 36, 65, or 67 etc. Alternatively, the number of intra-prediction modes may vary according to a block size or a color component type or both. For example, the number of intra prediction modes may vary according to whether the color component is a luma signal or a chroma signal. For example, as a block size becomes large, a number of intra-prediction modes may increase. Alternatively, a number of intra-prediction modes of a luma component block may be larger than a number of intra-prediction modes of a chroma component block.

An intra-prediction mode may be a non-angular mode or an angular mode. The non-angular mode may be a DC mode or a planar mode, and the angular mode may be a prediction mode having a specific direction or angle. The intra-prediction mode may be expressed by at least one of a mode number, a mode value, a mode numeral, a mode angle, and mode direction. A number of intra-prediction modes may be M, which is larger than 1, including the non-angular and the angular mode. In order to intra-predict a current block, a step of determining whether or not samples included in a reconstructed neighbor block may be used as reference samples of the current block may be performed. When a sample that is not usable as a reference sample of the current block is present, a value obtained by duplicating or performing interpolation on at least one sample value among samples included in the reconstructed neighbor block or both may be

used to replace with a non-usable sample value of a sample, thus the replaced sample value is used as a reference sample of the current block.

FIG. 7 is a diagram illustrating reference samples capable of being used for intra prediction.

As shown in FIG. 7, at least one of the reference sample line 0 to the reference sample line 3 may be used for intra prediction of the current block. In FIG. 7, the samples of a segment A and a segment F may be padded with the samples closest to a segment B and a segment E, respectively, instead of retrieving from the reconstructed neighboring block. Index information indicating the reference sample line to be used for intra prediction of the current block may be signaled. When the upper boundary of the current block is the boundary of the CTU, only the reference sample line 0 may be available. Therefore, in this case, the index information may not be signaled. When a reference sample line other than the reference sample line 0 is used, filtering for a prediction block, which will be described later, may not be performed.

When intra-predicting, a filter may be applied to at least one of a reference sample and a prediction sample based on an intra-prediction mode and a current block size.

In case of a planar mode, when generating a prediction block of a current block, according to a position of a prediction target sample within a prediction block, a sample value of the prediction target sample may be generated by using a weighted sum of an upper and left side reference sample of a current sample, and a right upper side and left lower side reference sample of the current block. In addition, in case of a DC mode, when generating a prediction block of a current block, an average value of upper side and left side reference samples of the current block may be used. In addition, in case of an angular mode, a prediction block may be generated by using an upper side, a left side, a right upper side, and/or a left lower side reference sample of the current block. In order to generate a prediction sample value, interpolation of a real number unit may be performed.

In the case of intra prediction between color components, a prediction block for the current block of the second color component may be generated on the basis of the corresponding reconstructed block of the first color component. For example, the first color component may be a luma component, and the second color component may be a chroma component. For intra prediction between color components, the parameters of the linear model between the first color component and the second color component may be derived on the basis of the template. The template may include upper and/or left neighboring samples of the current block and upper and/or left neighboring samples of the reconstructed block of the first color component corresponding thereto. For example, the parameters of the linear model may be derived using a sample value of a first color component having a maximum value among samples in a template and a sample value of a second color component corresponding thereto, and a sample value of a first color component having a minimum value among samples in the template and a sample value of a second color component corresponding thereto. When the parameters of the linear model are derived, a corresponding reconstructed block may be applied to the linear model to generate a prediction block for the current block. According to a video format, subsampling may be performed on the neighboring samples of the reconstructed block of the first color component and the corresponding reconstructed block. For example, when one sample of the second color component corresponds to four samples of the first color component, four samples of the

first color component may be sub-sampled to compute one corresponding sample. In this case, the parameter derivation of the linear model and intra prediction between color components may be performed on the basis of the corresponding sub-sampled samples. Whether or not to perform intra prediction between color components and/or the range of the template may be signaled as the intra prediction mode.

The current block may be partitioned into two or four sub-blocks in the horizontal or vertical direction. The partitioned sub-blocks may be sequentially reconstructed. That is, the intra prediction may be performed on the sub-block to generate the sub-prediction block. In addition, dequantization and/or inverse transform may be performed on the sub-blocks to generate sub-residual blocks. A reconstructed sub-block may be generated by adding the sub-prediction block to the sub-residual block. The reconstructed sub-block may be used as a reference sample for intra prediction of the sub-sub-blocks. The sub-block may be a block including a predetermined number (for example, 16) or more samples. Accordingly, for example, when the current block is an 8×4 block or a 4×8 block, the current block may be partitioned into two sub-blocks. Also, when the current block is a 4×4 block, the current block may not be partitioned into sub-blocks. When the current block has other sizes, the current block may be partitioned into four sub-blocks. Information on whether or not to perform the intra prediction based on the sub-blocks and/or the partitioning direction (horizontal or vertical) may be signaled. The intra prediction based on the sub-blocks may be limited to be performed only when reference sample line 0 is used. When the intra prediction based on the sub-block is performed, filtering for the prediction block, which will be described later, may not be performed.

The final prediction block may be generated by performing filtering on the prediction block that is intra-predicted. The filtering may be performed by applying predetermined weights to the filtering target sample, the left reference sample, the upper reference sample, and/or the upper left reference sample. The weight and/or the reference sample (range, position, etc.) used for the filtering may be determined on the basis of at least one of a block size, an intra prediction mode, and a position of the filtering target sample in the prediction block. The filtering may be performed only in the case of a predetermined intra prediction mode (e.g., DC, planar, vertical, horizontal, diagonal, and/or adjacent diagonal modes). The adjacent diagonal mode may be a mode in which k is added to or subtracted from the diagonal mode. For example, k may be a positive integer of 8 or less.

An intra-prediction mode of a current block may be entropy encoded/decoded by predicting an intra-prediction mode of a block present adjacent to the current block. When intra-prediction modes of the current block and the neighbor block are identical, information that the intra-prediction modes of the current block and the neighbor block are identical may be signaled by using predetermined flag information. In addition, indicator information of an intra-prediction mode that is identical to the intra-prediction mode of the current block among intra-prediction modes of a plurality of neighbor blocks may be signaled. When intra-prediction modes of the current block and the neighbor block are different, intra-prediction mode information of the current block may be entropy encoded/decoded by performing entropy encoding/decoding based on the intra-prediction mode of the neighbor block.

FIG. 5 is a diagram illustrating an embodiment of an inter-picture prediction process.

In FIG. 5, an arrow represents a prediction direction. Pictures may be categorized into intra pictures (I pictures), predictive pictures (P pictures), and Bi-predictive pictures (B pictures) according to the encoding type thereof.

The I picture may be encoded through intra-prediction without requiring inter-picture prediction. The P picture may be encoded through inter-picture prediction by using a reference picture that is present in one direction (i.e., forward direction or backward direction) with respect to a current block. The B picture may be encoded through inter-picture prediction by using reference pictures that are preset in two directions (i.e., forward direction and backward direction) with respect to a current block. When the inter-picture prediction is used, the encoder may perform inter-picture prediction or motion compensation and the decoder may perform the corresponding motion compensation.

Hereinbelow, an embodiment of the inter-picture prediction will be described in detail.

The inter-picture prediction or motion compensation may be performed using a reference picture and motion information.

Motion information of a current block may be derived during inter-picture prediction by each of the encoding apparatus 100 and the decoding apparatus 200. The motion information of the current block may be derived by using motion information of a reconstructed neighboring block, motion information of a collocated block (also referred to as a col block or a co-located block), and/or a block adjacent to the co-located block. The co-located block may mean a block that is located spatially at the same position as the current block, within a previously reconstructed collocated picture (also referred to as a col picture or a co-located picture). The co-located picture may be one picture among one or more reference pictures included in a reference picture list.

The derivation method of the motion information may be different depending on the prediction mode of the current block. For example, a prediction mode applied for inter prediction includes an AMVP mode, a merge mode, a skip mode, a merge mode with a motion vector difference, a subblock merge mode, a triangle partitioning mode, an combined inter intra prediction mode, affine mode, and the like. Herein, the merge mode may be referred to as a motion merge mode.

For example, when the AMVP is used as the prediction mode, at least one of motion vectors of the reconstructed neighboring blocks, motion vectors of the co-located blocks, motion vectors of blocks adjacent to the co-located blocks, and a (0, 0) motion vector may be determined as motion vector candidates for the current block, and a motion vector candidate list is generated by using the motion vector candidates. The motion vector candidate of the current block can be derived by using the generated motion vector candidate list. The motion information of the current block may be determined based on the derived motion vector candidate. The motion vectors of the collocated blocks or the motion vectors of the blocks adjacent to the collocated blocks may be referred to as temporal motion vector candidates, and the motion vectors of the reconstructed neighboring blocks may be referred to as spatial motion vector candidates.

The encoding apparatus 100 may calculate a motion vector difference (MVD) between the motion vector of the current block and the motion vector candidate and may

perform entropy encoding on the motion vector difference (MVD). In addition, the encoding apparatus **100** may perform entropy encoding on a motion vector candidate index and generate a bitstream. The motion vector candidate index may indicate an optimum motion vector candidate among the motion vector candidates included in the motion vector candidate list. The decoding apparatus may perform entropy decoding on the motion vector candidate index included in the bitstream and may select a motion vector candidate of a decoding target block from among the motion vector candidates included in the motion vector candidate list by using the entropy-decoded motion vector candidate index. In addition, the decoding apparatus **200** may add the entropy-decoded MVD and the motion vector candidate extracted through the entropy decoding, thereby deriving the motion vector of the decoding target block.

Meanwhile, the coding apparatus **100** may perform entropy-coding on resolution information of the calculated MVD. The decoding apparatus **200** may adjust the resolution of the entropy-decoded MVD using the MVD resolution information.

Meanwhile, the coding apparatus **100** calculates a motion vector difference (MVD) between a motion vector and a motion vector candidate in the current block on the basis of an affine model, and performs entropy-coding on the MVD. The decoding apparatus **200** derives a motion vector on a per sub-block basis by deriving an affine control motion vector of a decoding target block through the sum of the entropy-decoded MVD and an affine control motion vector candidate.

The bitstream may include a reference picture index indicating a reference picture. The reference picture index may be entropy-encoded by the encoding apparatus **100** and then signaled as a bitstream to the decoding apparatus **200**. The decoding apparatus **200** may generate a prediction block of the decoding target block based on the derived motion vector and the reference picture index information.

Another example of the method of deriving the motion information of the current may be the merge mode. The merge mode may mean a method of merging motion of a plurality of blocks. The merge mode may mean a mode of deriving the motion information of the current block from the motion information of the neighboring blocks. When the merge mode is applied, the merge candidate list may be generated using the motion information of the reconstructed neighboring blocks and/or the motion information of the collocated blocks. The motion information may include at least one of a motion vector, a reference picture index, and an inter-picture prediction indicator. The prediction indicator may indicate one-direction prediction (L0 prediction or L1 prediction) or two-direction predictions (L0 prediction and L1 prediction).

The merge candidate list may be a list of motion information stored. The motion information included in the merge candidate list may be at least one of motion information (spatial merge candidate) of a neighboring block adjacent to the current block, motion information (temporal merge candidate) of the collocated block of the current block in the reference picture, new motion information generated by a combination of the motion information existing in the merge candidate list, motion information (history-based merge candidate) of the block that is encoded/decoded before the current block, and zero merge candidate.

The encoding apparatus **100** may generate a bitstream by performing entropy encoding on at least one of a merge flag and a merge index and may signal the bitstream to the decoding apparatus **200**. The merge flag may be information

indicating whether or not to perform the merge mode for each block, and the merge index may be information indicating that which neighboring block, among the neighboring blocks of the current block, is a merge target block. For example, the neighboring blocks of the current block may include a left neighboring block on the left of the current block, an upper neighboring block disposed above the current block, and a temporal neighboring block temporally adjacent to the current block.

Meanwhile, the coding apparatus **100** performs entropy-coding on the correction information for correcting the motion vector among the motion information of the merge candidate and signals the same to the decoding apparatus **200**. The decoding apparatus **200** can correct the motion vector of the merge candidate selected by the merge index on the basis of the correction information. Here, the correction information may include at least one of information on whether or not to perform the correction, correction direction information, and correction size information. As described above, the prediction mode that corrects the motion vector of the merge candidate on the basis of the signaled correction information may be referred to as a merge mode having the motion vector difference.

The skip mode may be a mode in which the motion information of the neighboring block is applied to the current block as it is. When the skip mode is applied, the encoding apparatus **100** may perform entropy encoding on information of the fact that the motion information of which block is to be used as the motion information of the current block to generate a bit stream, and may signal the bitstream to the decoding apparatus **200**. The encoding apparatus **100** may not signal a syntax element regarding at least any one of the motion vector difference information, the encoding block flag, and the transform coefficient level to the decoding apparatus **200**.

The subblock merge mode may mean a mode that derives the motion information in units of sub-blocks of a coding block (CU). When the subblock merge mode is applied, a subblock merge candidate list may be generated using motion information (sub-block based temporal merge candidate) of the sub-block collocated to the current sub-block in the reference image and/or an affine control point motion vector merge candidate.

The triangle partitioning mode may mean a mode that derives motion information by partitioning the current block into diagonal directions, derives each prediction sample using each of the derived motion information, and derives the prediction sample of the current block by weighting each of the derived prediction samples.

The inter-intra combined prediction mode may mean a mode that derives a prediction sample of the current block by weighting a prediction sample generated by inter prediction and a prediction sample generated by intra prediction.

The decoding apparatus **200** may correct the derived motion information by itself. The decoding apparatus **200** may search the predetermined region on the basis of the reference block indicated by the derived motion information and derive the motion information having the minimum SAD as the corrected motion information.

The decoding apparatus **200** may compensate a prediction sample derived via inter prediction using an optical flow.

FIG. 6 is a diagram illustrating a transform and quantization process.

As illustrated in FIG. 6, a transform and/or quantization process is performed on a residual signal to generate a quantized level signal. The residual signal is a difference between an original block and a prediction block (i.e., an

intra prediction block or an inter prediction block). The prediction block is a block generated through intra prediction or inter prediction. The transform may be a primary transform, a secondary transform, or both. The primary transform of the residual signal results in transform coefficients, and the secondary transform of the transform coefficients results in secondary transform coefficients.

At least one scheme selected from among various transform schemes which are preliminarily defined is used to perform the primary transform. For example, examples of the predefined transform schemes include discrete cosine transform (DCT), discrete sine transform (DST), and Karhunen-Loève transform (KLT). The transform coefficients generated through the primary transform may undergo the secondary transform. The transform schemes used for the primary transform and/or the secondary transform may be determined according to coding parameters of the current block and/or neighboring blocks of the current block. Alternatively, transform information indicating the transform scheme may be signaled. The DCT-based transform may include, for example, DCT-2, DCT-8, and the like. The DST-based transform may include, for example, DST-7.

A quantized-level signal (quantization coefficients) may be generated by performing quantization on the residual signal or a result of performing the primary transform and/or the secondary transform. The quantized level signal may be scanned according to at least one of a diagonal up-right scan, a vertical scan, and a horizontal scan, depending on an intra prediction mode of a block or a block size/shape. For example, as the coefficients are scanned in a diagonal up-right scan, the coefficients in a block form change into a one-dimensional vector form. Aside from the diagonal up-right scan, the horizontal scan of horizontally scanning a two-dimensional block form of coefficients or the vertical scan of vertically scanning a two-dimensional block form of coefficients may be used depending on the intra prediction mode and/or the size of a transform block. The scanned quantized-level coefficients may be entropy-encoded to be inserted into a bitstream.

A decoder entropy-decodes the bitstream to obtain the quantized-level coefficients. The quantized-level coefficients may be arranged in a two-dimensional block form through inverse scanning. For the inverse scanning, at least one of a diagonal up-right scan, a vertical scan, and a horizontal scan may be used.

The quantized-level coefficients may then be dequantized, then be secondary-inverse-transformed as necessary, and finally be primary-inverse-transformed as necessary to generate a reconstructed residual signal.

Inverse mapping in a dynamic range may be performed for a luma component reconstructed through intra prediction or inter prediction before in-loop filtering. The dynamic range may be divided into 16 equal pieces and the mapping function for each piece may be signaled. The mapping function may be signaled at a slice level or a tile group level. An inverse mapping function for performing the inverse mapping may be derived on the basis of the mapping function. In-loop filtering, reference picture storage, and motion compensation are performed in an inverse mapped region, and a prediction block generated through inter prediction is converted into a mapped region via mapping using the mapping function, and then used for generating the reconstructed block. However, since the intra prediction is performed in the mapped region, the prediction block generated via the intra prediction may be used for generating the reconstructed block without mapping/inverse mapping.

When the current block is a residual block of a chroma component, the residual block may be converted into an inverse mapped region by performing scaling on the chroma component of the mapped region. The availability of the scaling may be signaled at the slice level or the tile group level. The scaling may be applied only when the mapping for the luma component is available and the division of the luma component and the division of the chroma component follow the same tree structure. The scaling may be performed on the basis of an average of sample values of a luma prediction block corresponding to the color difference block. In this case, when the current block uses inter prediction, the luma prediction block may mean a mapped luma prediction block. A value necessary for the scaling may be derived by referring to a lookup table using an index of a piece to which an average of sample values of a luma prediction block belongs. Finally, by scaling the residual block using the derived value, the residual block may be switched to the inverse mapped region. Then, chroma component block restoration, intra prediction, inter prediction, in-loop filtering, and reference picture storage may be performed in the inverse mapped area.

Information indicating whether the mapping/inverse mapping of the luma component and chroma component is available may be signaled through a set of sequence parameters.

The prediction block of the current block may be generated on the basis of a block vector indicating a displacement between the current block and the reference block in the current picture. In this way, a prediction mode for generating a prediction block with reference to the current picture is referred to as an intra block copy (IBC) mode. The IBC mode may be applied to  $M \times N$  ( $M \leq 64$ ,  $N \leq 64$ ) coding units. The IBC mode may include a skip mode, a merge mode, an AMVP mode, and the like. In the case of a skip mode or a merge mode, a merge candidate list is constructed, and the merge index is signaled so that one merge candidate may be specified. The block vector of the specified merge candidate may be used as a block vector of the current block. The merge candidate list may include at least one of a spatial candidate, a history-based candidate, a candidate based on an average of two candidates, and a zero-merge candidate. In the case of an AMVP mode, the difference block vector may be signaled. In addition, the prediction block vector may be derived from the left neighboring block and the upper neighboring block of the current block. The index on which neighboring block to use may be signaled. The prediction block in the IBC mode is included in the current CTU or the left CTU and limited to a block in the already reconstructed area. For example, a value of the block vector may be limited such that the prediction block of the current block is positioned in an area of three  $64 \times 64$  blocks preceding the  $64 \times 64$  block to which the current block belongs in the coding/decoding order. By limiting the value of the block vector in this way, memory consumption and device complexity according to the IBC mode implementation may be reduced.

Hereinafter, embodiments of the present invention will be described with reference to FIG. 8 to FIG. 22.

For efficient encoding/decoding of an image, a picture may be partitioned into one or more slices. Herein, each slice may have an arbitrary shape (for example, square and/or rectangle) and include one or more coding tree units (CTUs).

In addition, a picture and/or a slice may be partitioned into one or more tiles. Herein, each tile may have an arbitrary shape (for example, square and/or rectangle) and include one or more coding tree units (CTUs). CTUs belonging to a tile

may be scanned in a raster scan order inside the tile. In addition, a slice may be used in the same meaning as a tile group. A tile may consist of a plurality of slices. Herein, a slice may be understood in the same meaning as a brick described below. Only through vertical partitioning, a single tile may be divided into a plurality of slices. Alternatively, only through horizontal partitioning, a single tile may be divided into a plurality of slices. A partitioning direction may be a fixed direction that is preset in an encoding/decoding apparatus or be adaptively determined based on a flag indicating it.

A flag indicating whether or not a single picture is partitioned into a plurality of tiles or slices may be used. For example, when the flag has a first value, a current picture may not be partitioned into a plurality of tiles or slices. This may mean that a current picture consists of one tile or one slice. On the other hand, when the flag has a second value, a current picture may be partitioned into a plurality of tiles or slices.

Only when a current picture is partitioned into a plurality of regions (for example, tiles and slices) on the basis of a value of the flag, at least one of tile partitioning information and slice partitioning information may be signaled.

Herein, tile partitioning information may include at least one of information on the number of tile widths and information on the number of tile heights. Information on the number of tile widths may indicate the number of tile columns constituting a single picture, and information on the number of tile heights may indicate the number of tile rows constituting a single picture. Depending on information on the number of tile widths or heights, width information for each tile column and height information for each tile row may be signaled respectively. In other words, tile partitioning information may further include at least one of width information and height information. Width information and height information of a tile may be signaled irrespective of whether or not a size of tile constituting a picture is uniform. Width information and height information of a tile may be encoded in a unit that is predefined in an encoding apparatus. Herein, a predefined unit may be a coding tree block (CTB), a CTB row, and n-sample grid (Here, n may be 4, 8 or 16).

The slice partitioning information may be used to partition a single picture into a plurality of slices. Slice partitioning information may include information on the number of slices constituting a single picture and information on the width or height of each slice. Width information and height information of a slice may be encoded in a unit that is predefined in an encoding apparatus. Herein, a predefined unit may be a tile, a coding tree block (CTB), a CTB row, and n-sample grid (Here, n may be 4, 8 or 16).

As described above, a single tile may consist of a plurality of slices. In this case, slice partitioning information may include at least one of information on the number of slices constituting a single slice and height information or width information of each slice. Both height information and width information may be signaled. Alternatively, either height information or width information may be signaled. Width information or height information may be encoded in a unit that is predefined in an encoding apparatus. Herein, a predefined unit may be a coding tree block (CTB), a CTB row, and n-sample grid (Here, n may be 4, 8 or 16). Meanwhile, a unit for encoding the size information of a tile and a unit for encoding the size information of a slice may be different from each other. A unit for encoding the size information of slice, when a plurality of tiles is included in

one slice, may be different from a unit for encoding the size information of slice, when a plurality of tiles is included in one tile.

Based on whether or not a plurality of tiles included in a slice has a raster scan order (hereinafter, referred to as 'Condition 1'), slice partitioning information may be selectively signaled. For example, when the condition 1 is satisfied, slice partitioning information may not be signaled. When the condition 1 is not satisfied, slice partitioning information may be signaled. Alternatively, based on whether or not a plurality of tiles included in a slice covers a rectangular region in a picture (hereinafter, referred to as 'Condition 2'), slice partitioning information may be selectively signaled. For example, when the condition 2 is satisfied, slice partitioning information may not be signaled. When the condition 2 is not satisfied, slice partitioning information may be signaled. A separate flag may be used to determine at least one of the above-described condition 1 and condition 2. The flag may be encoded and signaled in an encoding apparatus and be adaptively determined based on the above-described coding parameter.

More particularly, a slice (or tile group) may include a tile in a raster scan order inside a picture. Alternatively, a slice (or tile group) may be configured as a rectangular region in a picture. Information on whether a slice (or tile group) constituting a picture includes a tile in a raster order or is configured as a rectangular region in a picture may be signaled as slice (or tile group) mode information. Herein, slice (or tile group) mode information may be signaled at an upper level, for example, in a picture parameter set or a sequence parameter set.

For example, information on whether or not a slice constituting a picture is configured as a rectangular region is slice mode information. The information may be signaled by a rectangular slice indicator (for example, `rect_slice_flag`). Herein, when the rectangular slice indicator has a first value (for example, 1), a slice may be configured as a rectangular region in a picture. Alternatively, when the rectangular slice indicator has a second value (for example, 0), a slice may include a tile in a raster scan order.

When a slice includes at least one tile, the at least one tile may consist of at least one brick. Here, the at least one brick may consist of CTU rows constituting a tile.

Alternatively, when the at least one tile is not partitioned into bricks, the tile may be considered as a single brick.

When a slice includes a plurality of bricks, each brick may be encoded/decoded in a raster scan order inside a picture or the slice including the bricks.

At least one of the position, size and shape of a slice may be determined based on the positional information of a brick. The positional information of the brick may be defined in an index form (for example, `idx=0, 1, . . . , n-1`) specifying the brick. Alternatively, the positional information may be defined by a coordinate of the brick (for example, a coordinate of at least one of the top left, top right, bottom left and bottom right samples of the brick). There may be one or more pieces of the positional information of the brick. For example, one slice may be specified based on the positional information of a top left brick and/or that of a bottom right brick inside the slice.

Every positional information of the brick may be encoded and signaled in an encoder. Alternatively, some of the positional information of the brick may be encoded and signaled, and the remaining positional information of the brick may be derived in a decoder. Alternatively, every positional information of the brick may be derived in a decoder.

The derivation may be performed by considering the position of a slice to which a corresponding brick belongs. For example, whether or not a slice is located in a corner region inside a picture may be considered. Herein, a corner region may include at least one of a top left region, a top right region, a bottom left region, and a bottom right region. When a slice to which a corresponding brick belongs is located in a corner region inside a picture, only some of the positional information of the brick may be signaled.

In addition, the derivation may be performed by considering whether or not a slice adjoins a boundary of a picture. Here, the boundary may mean at least one of the left, right, top and bottom of the picture. Alternatively, only whether or not a slice adjoins a specific boundary among the boundaries may be considered. For example, when a current slice adjoins the right and bottom boundaries of a picture, the positional information of a top left brick may be signaled and the positional information of a bottom right brick may be derived in a decoder. As another example, when a current slice adjoins the left and top boundaries of a picture, the positional information of a bottom right brick may be signaled and the positional information of a top left brick may be derived in a decoder.

In addition, the derivation may be performed based on the number of slices (for example, M) belonging to a picture. For example, when a current slice is the i-th slice and i is equal to M that is the number of slices, the positional information of a lower-right brick may not be signaled but be derived in a decoder. Alternatively, when i is not equal to M, the positional information of a top left brick and the positional information of a bottom right brick may be signaled respectively.

The positional information of a brick belonging to a current slice may be derived based on at least one of a plurality of pieces of information on bricks used in a previous slice. Herein, a previous slice may mean a slice adjacent to at least one of the left, top, top left, bottom left and bottom right of a current slice.

For example, a difference value or an offset between the positional information of a top left brick of a current slice and that of a top left brick of a previous slice may be signaled. In this case, the positional information of the top left brick of the current slice may be derived based on the positional information of the top left brick of the previous slice and the signaled difference value. Alternatively, the positional information of the top left brick of the current slice may be derived by using the positional information of the top left brick and that of the bottom right brick of the previous slice.

FIG. 8 is a view illustrating an example of signaling brick-related information according to an embodiment of the present invention.

FIG. 9 is a view for explaining an example of brick-related information according to an embodiment of the present invention.

Partitioning information of a tile, a brick and/or a slice for a current picture may be transmitted inside a picture parameter set (PPS). For example, referring to FIG. 8, information indicating whether or not a slice constituting a current picture is configured as a rectangular region (for example, `rect_slice_flag`), information indicating whether or not each slice is configured as a single brick (for example, `single_brick_per_slice_flag`) and/or information indicating the number of slices in a picture (for example, `num_slices_in_pic_minus1`) may be transmitted. In addition, as index information of a brick included in each slice (for example, i-th slice), index information of a top left brick (for example,

`top_left_brick_idx[i]`) and/or delta index information of a bottom right brick (for example, `bottom_right_brick_idx_delta[i]`) may be transmitted.

More particularly, referring to FIG. 8, information indicating whether or not each slice in a picture is configured as a single brick (for example, `single_brick_per_slice_flag`) may be signaled within a picture parameter set. For example, when `single_brick_per_slice_flag` has a first value (for example, 1), the slice may be configured as a single brick. In other words, in a picture, the number of slices may be identical with that of bricks. Alternatively, on the contrary, when `single_brick_per_slice_flag` has a second value (for example, 0), the slice may not be configured as a single brick.

Herein, when each slice is not configured as a single brick (for example, `single_brick_per_slice_flag` has a value of 0), a rectangular slice indicator (for example, `rect_slice_flag`) indicating whether or not a slice constituting a current picture is configured as a rectangular region may be signaled. Herein, when the rectangular slice indicator has a first value (for example, 1), a slice may be configured as a rectangular region in a picture. Alternatively, when the rectangular slice indicator has a second value (for example, 0), a slice may include a tile in a raster scan order.

In addition, when a slice in a current picture is configured as a rectangular region (for example, `rect_slice_flag` has a value of 1) and each slice is not configured as a single brick (for example, `single_brick_per_slice_flag` has a value of 0), information indicating the number of slices in a picture (for example, `num_slices_in_pic_minus1`) may be signaled. Herein, among a plurality of slices existing in a current picture, a position of a slice according to a raster scan order may be represented by 'i'. The i may be a positive integer from 0 to `num_slices_in_pic_minus1`.

In addition, for each slice constituting a picture (for example, i-th slice), index information of a top left brick (for example, `top_left_brick_idx[i]`) included in the slice and delta index information of a bottom right brick (for example, `bottom_right_brick_idx_delta[i]`) may be signaled respectively.

Herein, referring to FIG. 9, the delta index information of the bottom right brick (for example, `bottom_right_brick_idx_delta[i]`) may be information indicating a difference between index information of a bottom right brick (for example, `botRightBkIdx`) at the bottom right corner of each slice (for example, i-th slice) and the index information of the top left brick (for example, `top_left_brick_idx[i]`). In other words, index information of a bottom right brick (for example, `botRightBkIdx`) may be determined based on index information of a top left brick (for example, `top_left_brick_idx[i]`). In other words, index information of a bottom right brick may be determined as a sum of index information of a top left brick and delta index information of the bottom right brick. Alternatively, index information of a top left brick (for example, `top_left_brick_idx[i]`) may be determined based on index information of a bottom right brick.

FIG. 10 is a view illustrating another example of signaling brick-related information according to an embodiment of the present invention.

FIG. 11 is a view for explaining another example of brick-related information according to an embodiment of the present invention.

Referring to FIG. 10, unlike the above-described embodiment related to FIG. 8, in the case of a last slice existing in a picture, signaling of delta index information of a bottom right brick (for example, `bottom_right_brick_idx_delta[i]`) may be skipped.

35

For example, when 'i' indicating the position of a slice in a current picture has a same value as information indicating the number of slices existing in a current picture (`num_slices_in_pic_minus1`), that is, when a corresponding slice is a last slice in a current picture, signaling of delta index information of a bottom right brick (`bottom_right_brick_idx_delta[i]`) may be skipped.

In addition, referring to FIG. 11, in the case of a last slice existing in a current picture (for example, when 'i' is identical with `num_slices_in_pic_minus1`), since signaling of delta index information of a bottom right brick (for example, `bottom_right_brick_idx_delta[i]`) is skipped, index information of a bottom right brick (for example, `botRightBkIdx`) may be determined by applying a predetermined offset to the number of bricks existing in a picture (for example, `NumBricksInPic`). For example, index information of a bottom right brick (`botRightBkIdx`) may be determined as the number of bricks in a picture `NumBricksInPic-1`.

FIG. 12 is a view illustrating another example of signaling brick-related information according to an embodiment of the present invention.

FIG. 13 and FIG. 14 are views for explaining another example of brick-related information according to an embodiment of the present invention.

When partitioning information of a tile, a brick, and a slice for a current picture is transmitted inside a picture parameter set (PPS), information on the number of bricks existing in each tile may be signaled.

For example, referring to FIG. 12, information indicating whether or not at least one tile existing in a picture is partitioned into bricks (for example, `brick_splitting_present_flag`), for every tile in a picture, information indicating whether or not an i-th tile is partitioned into bricks (for example, `brick_split_flag[i]`), information indicating whether or not vertical sizes of bricks included in the i-th tile are uniformly partitioned (for example, `uniform_brick_spacing_flag[i]`), and/or information indicating the number of split bricks included in the i-th tile (for example, `num_brick_rows_minus1[i]`) may be transmitted.

Herein, referring to FIGS. 12 to 14, information indicating the number of bricks included in the i-th tile (for example, `num_brick_rows_minus1[i]`) may mean a value that is less than the actual number of bricks included in an i-th tile by 1. In other words, a value obtained by adding 1 to the transmitted information indicating the number of bricks included in the i-th tile (for example, `num_brick_rows_minus1[i]`) may be the number of bricks in the i-th tile. Here, `num_brick_rows_minus1[i]` may have a value from 1 to `RowHeight[i]-1`. In addition, when an i-th tile is not partitioned into bricks (for example, `brick_split_flag[i]` is 0), `num_brick_rows_minus1[i]` may be derived as 0.

More particularly, when at least one tile existing in a current picture is partitioned into bricks, flag information may be transmitted. For example, when `brick_splitting_present_flag` has a first value (for example, 1), at least one tile existing in a current picture may be partitioned into bricks. In addition, when a tile in a picture (for example, i-th tile) is partitioned into bricks (for example, `brick_split_flag[i]` is 1) and a vertical size of bricks included in a corresponding tile is not uniformly partitioned (for example, `uniform_brick_spacing_flag[i]` is 0), information indicating the number of partitioned bricks in a corresponding tile (for example, i-th tile) may be transmitted (for example, `num_brick_rows_minus1[i]`).

FIG. 15 is a view illustrating another example of signaling brick-related information according to an embodiment of the present invention.

36

FIG. 16 and FIG. 17 are views for explaining another example of brick-related information according to an embodiment of the present invention.

When partitioning information of a tile, a brick, and a slice for a current picture is transmitted inside a picture parameter set (PPS), information on the number of bricks existing in each tile may be signaled.

For example, referring to FIG. 12, information indicating whether or not at least one tile existing in a picture is partitioned into bricks (for example, `brick_splitting_present_flag`), for every tile in a picture, information indicating whether or not an i-th tile is partitioned into bricks (for example, `brick_split_flag[i]`), information indicating whether or not vertical sizes of bricks included in the i-th tile are uniformly partitioned (for example, `uniform_brick_spacing_flag[i]`), and/or information indicating the number of split bricks included in the i-th tile (for example, `num_brick_rows_minus2[i]`) may be transmitted.

Herein, referring to FIGS. 15 to 17, unlike the above-described embodiment related to FIGS. 12 to 14, in the present embodiment, information indicating the number of partitioned bricks in an i-th tile (for example, `num_brick_rows_minus2[i]`) may mean a value that is less than the actual number of bricks included in the i-th tile by 2. In other words, a value obtained by adding 2 to the transmitted information indicating the number of partitioned bricks in the i-th tile (for example, `num_brick_rows_minus2[i]`) may be the number of bricks in the i-th tile. Here, `num_brick_rows_minus2[i]` may have a value from 0 to `RowHeight[i]-2`. In addition, when an i-th tile is not partitioned into bricks (for example, `brick_split_flag[i]` is 0), `num_brick_rows_minus2[i]` may be derived as 0.

More particularly, when at least one tile existing in a current picture is partitioned into bricks, flag information may be transmitted. For example, when `brick_splitting_present_flag` has a first value (for example, 1), at least one tile existing in a current picture may be partitioned into bricks. In addition, when a tile in a picture (for example, i-th tile) is partitioned into bricks (for example, `brick_split_flag[i]` is 1) and a vertical size of bricks included in a corresponding tile is not uniformly partitioned (for example, `uniform_brick_spacing_flag[i]` is 0), information indicating the number of partitioned bricks in a corresponding tile (for example, i-th tile) may be transmitted (for example, `num_brick_rows_minus2[i]`).

FIG. 18 is a view illustrating an example of signaling for distinguishing a rectangular slice within a picture according to an embodiment of the present invention.

In the present specification, a rectangular slice may mean that a slice constituting a picture is configured as a rectangular region. Herein, as described above, information on whether or not a slice constituting a picture is configured as a rectangular region is slice mode information. The information may be signaled by a rectangular slice indicator (for example, `rect_slice_flag`). In other words, a description on a rectangular slice may assume that the rectangular slice indicator (for example, `rect_slice_flag`) has a specific value (for example, 1).

According to an embodiment, a picture may be partitioned into tiles, and a single tile may be partitioned into a plurality of rectangular slices.

Alternatively, a picture may be partitioned into tiles, and one or more tiles may constitute a single rectangular slice.

An embodiment may be a method of distinguishing a rectangular slice in a picture.

According to an embodiment, when a slice constituting a picture is a rectangular slice (for example, `rect_slice_flag` is

1), information related to the number of slices in a current picture may be signaled. For example, `num_slices_in_pic_minus1`, which corresponds to (the number of slices in a picture-1), may be signaled. In addition, a rectangular slice may be configured by performing iteration as many times as the number of slices existing in a picture.

In this case, based on information related to the number of slices in the current picture (for example, `num_slices_in_pic_minus1`), tile information related to the slice may be signaled. For example, when `num_slices_in_pic_minus1` is greater than N, tile information related to the slice may be signaled. Here, N may be 1.

In addition, tile information related to the slice may include information related to a tile index and information indicating whether or not the slice is capable of being specified by the information related to a tile index. Here, the information related to a tile index may be a difference value of the tile index (for example, `tile_idx_delta[i]`).

According to an embodiment, after a rectangular slice (for example, an *i*-th slice) is configured, a difference value of a tile index (for example, `tile_idx_delta[i]`) may be transmitted and thus a first initial tile of a next rectangular slice may be determined.

Herein, the difference value of a tile index (for example, `tile_idx_delta[i]`) may be a difference value between a tile index including a first coding tree unit (CTU) in a (*i*+1)-th rectangular slice and a tile index including a first CTU in an *i*-th rectangular slice.

FIG. 19 is a view illustrating another example of signaling for distinguishing a rectangular slice within a picture according to an embodiment of the present invention.

According to an embodiment, a picture may be partitioned into tiles, and a single tile may be partitioned into a plurality of rectangular slices.

Alternatively, a picture may be partitioned into tiles, and one or more tiles may constitute a single rectangular slice.

An embodiment may be a method of distinguishing a rectangular slice in a picture.

According to an embodiment, when a slice constituting a picture is a rectangular slice (for example, `rect_slice_flag` is 1), information related to the number of slices in a current picture may be signaled. For example, `num_slices_in_pic_minus1`, which corresponds to (the number of slices in a picture-1), may be signaled. In addition, a rectangular slice may be configured by performing iteration as many times as the number of slices existing in a picture.

In this case, based on information related to the number of slices in the current picture (for example, `num_slices_in_pic_minus1`), tile information related to the slice may be signaled. For example, when `num_slices_in_pic_minus1` is greater than N, tile information related to the slice may be signaled. Here, N may be 1.

In addition, tile information related to the slice may include information related to a tile index and information indicating whether or not the slice is capable of being specified by the information related to a tile index. Here, the information related to a tile index may be a difference value of the tile index (for example, `tile_idx_delta[i]`).

According to an embodiment, a difference value of a tile index (for example, `tile_idx_delta[i]`) may not be transmitted while defining a rectangular slice (for example, an *i*-th slice) but be transmitted from a next rectangular slice (for example, an (*i*+1)-th slice). Thus, a position of a slice in a picture may be specified.

FIG. 20 is a view illustrating another example of signaling for distinguishing a rectangular slice within a picture according to an embodiment of the present invention.

According to an embodiment, a picture may be partitioned into tiles, and a single tile may be partitioned into a plurality of rectangular slices.

Alternatively, a picture may be partitioned into tiles, and one or more tiles may constitute a single rectangular slice.

An embodiment may be a method of distinguishing a rectangular slice in a picture.

According to an embodiment, when a slice constituting a picture is a rectangular slice (for example, `rect_slice_flag` is 1), information related to the number of slices in a current picture may be signaled. For example, `num_slices_in_pic_minus1`, which corresponds to (the number of slices in a picture-1), may be signaled. In addition, a rectangular slice may be configured by performing iteration as many times as the number of slices existing in a picture.

In this case, based on information related to the number of slices in the current picture (for example, `num_slices_in_pic_minus1`), tile information related to the slice may be signaled. For example, when `num_slices_in_pic_minus1` is greater than N, tile information related to the slice may be signaled. Here, N may be 1.

In addition, tile information related to the slice may include information related to a tile index and information indicating whether or not the slice is capable of being specified by the information related to a tile index. Here, the information related to a tile index may be a difference value of the tile index (for example, `tile_idx_delta[i]`).

According to an embodiment, after a rectangular slice (for example, an *i*-th slice) is configured, a difference value of a tile index (for example, `tile_idx_delta[i]`) may be transmitted and thus a first initial tile of a next rectangular slice may be determined.

Herein, when a last rectangular slice (for example, an *i*-th slice) is a slice obtained by partitioning a single tile into a plurality of rectangular slices, a difference value of a tile index (for example, `tile_idx_delta[i]`) may not be transmitted and thus the amount of bits to be transmitted may be reduced.

FIG. 21 is a view for explaining an image decoding method according an embodiment of the present invention.

Referring to FIG. 21, an image decoder may derive slice mode information for at least one slice included in a current picture from a bitstream (S2101).

Meanwhile, slice mode information may indicate whether at least one slice included in a current picture is in a first slice mode consisting of tiles according to a raster scan order or in a second slice mode consisting of rectangular regions.

In addition, an image decoder may derive information on the number of slices on the basis of slice mode information (S2102). Herein, when the slice mode information is a second slice mode, the step S2102 may derive the information on the number of slices related to at least one slice included in a current picture.

In addition, an image decoder may derive tile information of a current slice on the basis of the information on the number of slices (S2103).

Meanwhile, slice mode information, information on the number of slices, and tile information may be signaled through a picture parameter set.

Meanwhile, tile information may include information on tile index presence and information on tile index difference value, and the information on tile index presence may indicate whether or not at least one slice included in a current picture is specified by the information on tile index difference value.

In addition, information on tile index difference value may indicate a difference value between a tile index of a



slice including a first CTU in a current slice and a tile index of a slice including a CTU in a next slice of the current slice.

Meanwhile, information on tile index difference value may not be signaled for a last slice included in a current picture.

In addition, an image decoder may decode a current slice on the basis of tile information (S2104).

FIG. 22 is a view for explaining an image encoding method according to an embodiment of the present invention.

Referring to FIG. 22, an image encoder may determine a slice mode related to at least one slice included in a current picture and encode slice mode information indicating the determined slice mode (S2201).

Meanwhile, slice mode information may indicate whether at least one slice included in a current picture is in a first slice mode consisting of tiles according to a raster scan order or in a second slice mode consisting of rectangular regions.

In addition, an encoder may determine information on the number of slices on the basis of a slice mode and encode the information on the number of slices (S2202). Herein, when the slice mode information is a second slice mode, the step S2202 may determine information on the number of slices, which is related to at least one slice included in a current picture.

In addition, an image encoder may determine tile information of a current slice on the basis of the information on the number of slices and encode tile information for a slice (S2203).

Meanwhile, slice mode information, information on the number of slices, and tile information may be signaled through a picture parameter set.

Meanwhile, tile information may include information on tile index presence and information on tile index difference value, and the information on tile index presence may indicate whether or not at least one slice included in a current picture is specified by the information on tile index difference value.

In addition, information on tile index difference value may indicate a difference value between a tile index of a slice including a first CTU in a current slice and a tile index of a slice including a CTU in a next slice of the current slice.

Meanwhile, information on tile index difference value may not be signaled for a last slice included in a current picture.

In addition, an image encoder may encode a current slice on the basis of tile information (S2204).

A bitstream generated by the image encoding method of the present invention may be temporarily stored in a computer readable non-transitory recording medium and may be decoded by the above-mentioned image decoding method.

More particularly, in a non-transitory computer-readable recording medium including a bitstream that is encoded by an image encoding method, the image encoding method may include: determining a slice mode related to at least one slice included in a current picture and encoding slice mode information indicating the determined slice mode; based on the slice mode, determining information on the number of slices and encoding the determined information on the number of slices; based on the information on the number of slices, determining tile information for a current slice and encoding the tile information for the current slice; and, based on the tile information, encoding the current slice. Herein, the tile information may include tile index presence information and tile index difference value information. The tile index difference value information may not be signaled for a last slice include in the current picture.

The above embodiments may be performed in the same method in an encoder and a decoder.

At least one or a combination of the above embodiments may be used to encode/decode a video.

A sequence of applying to above embodiment may be different between an encoder and a decoder, or the sequence applying to above embodiment may be the same in the encoder and the decoder.

The above embodiment may be performed on each luma signal and chroma signal, or the above embodiment may be identically performed on luma and chroma signals.

A block form to which the above embodiments of the present invention are applied may have a square form or a non-square form.

The above embodiment of the present invention may be applied depending on a size of at least one of a coding block, a prediction block, a transform block, a block, a current block, a coding unit, a prediction unit, a transform unit, a unit, and a current unit. Herein, the size may be defined as a minimum size or maximum size or both so that the above embodiments are applied, or may be defined as a fixed size to which the above embodiment is applied. In addition, in the above embodiments, a first embodiment may be applied to a first size, and a second embodiment may be applied to a second size. In other words, the above embodiments may be applied in combination depending on a size. In addition, the above embodiments may be applied when a size is equal to or greater than a minimum size and equal to or smaller than a maximum size. In other words, the above embodiments may be applied when a block size is included within a certain range.

For example, the above embodiments may be applied when a size of current block is 8×8 or greater. For example, the above embodiments may be applied when a size of current block is 4×4 only. For example, the above embodiments may be applied when a size of current block is 16×16 or smaller. For example, the above embodiments may be applied when a size of current block is equal to or greater than 16×16 and equal to or smaller than 64×64.

The above embodiments of the present invention may be applied depending on a temporal layer. In order to identify a temporal layer to which the above embodiments may be applied, a corresponding identifier may be signaled, and the above embodiments may be applied to a specified temporal layer identified by the corresponding identifier. Herein, the identifier may be defined as the lowest layer or the highest layer or both to which the above embodiment may be applied, or may be defined to indicate a specific layer to which the embodiment is applied. In addition, a fixed temporal layer to which the embodiment is applied may be defined.

For example, the above embodiments may be applied when a temporal layer of a current image is the lowest layer. For example, the above embodiments may be applied when a temporal layer identifier of a current image is 1. For example, the above embodiments may be applied when a temporal layer of a current image is the highest layer.

A slice type or a tile group type to which the above embodiments of the present invention are applied may be defined, and the above embodiments may be applied depending on the corresponding slice type or tile group type.

In the above-described embodiments, the methods are described based on the flowcharts with a series of steps or units, but the present invention is not limited to the order of the steps, and rather, some steps may be performed simultaneously or in different order with other steps. In addition, it should be appreciated by one of ordinary skill in the art

41

that the steps in the flowcharts do not exclude each other and that other steps may be added to the flowcharts or some of the steps may be deleted from the flowcharts without influencing the scope of the present invention.

The embodiments include various aspects of examples. All possible combinations for various aspects may not be described, but those skilled in the art will be able to recognize different combinations. Accordingly, the present invention may include all replacements, modifications, and changes within the scope of the claims.

The embodiments of the present invention may be implemented in a form of program instructions, which are executable by various computer components, and recorded in a computer-readable recording medium. The computer-readable recording medium may include stand-alone or a combination of program instructions, data files, data structures, etc. The program instructions recorded in the computer-readable recording medium may be specially designed and constructed for the present invention, or well-known to a person of ordinary skilled in computer software technology field. Examples of the computer-readable recording medium include magnetic recording media such as hard disks, floppy disks, and magnetic tapes; optical data storage media such as CD-ROMs or DVD-ROMs; magneto-optimum media such as floptical disks; and hardware devices, such as read-only memory (ROM), random-access memory (RAM), flash memory, etc., which are particularly structured to store and implement the program instruction. Examples of the program instructions include not only a mechanical language code formatted by a compiler but also a high level language code that may be implemented by a computer using an interpreter. The hardware devices may be configured to be operated by one or more software modules or vice versa to conduct the processes according to the present invention.

Although the present invention has been described in terms of specific items such as detailed elements as well as the limited embodiments and the drawings, they are only provided to help more general understanding of the invention, and the present invention is not limited to the above embodiments. It will be appreciated by those skilled in the art to which the present invention pertains that various modifications and changes may be made from the above description.

Therefore, the spirit of the present invention shall not be limited to the above-described embodiments, and the entire scope of the appended claims and their equivalents will fall within the scope and spirit of the invention.

#### INDUSTRIAL APPLICABILITY

The present invention may be used for an apparatus for encoding/decoding an image.

The invention claimed is:

1. An image decoding method, the method comprising:

decoding slice mode information, the slice mode information representing which one of a rectangular slice mode and a raster scan order mode is applied to a current picture,

wherein under the rectangular slice mode, all of slices in the current picture are determined as a rectangular shape, and

wherein under the raster scan order mode, at least one slice in the current picture has a shape other than the rectangular shape;

in response to the slice mode information indicating that the rectangular slice mode is applied, decoding slice number information, the slice number information rep-

42

resenting a value subtracting 1 from a number of slices included in the current picture; and

decoding tile index difference value information for a first slice in the current picture, the first slice being composed of one or more tile in the current picture,

wherein the tile index difference value information represents a difference between an index of a first tile in the first slice and an index of a first tile in a second slice in the current picture,

wherein the index of the first tile in the second slice is derived by the tile index difference value information, and

wherein the tile index difference value information is decoded from a bitstream in response to the slice number information being greater than a pre-defined value.

2. The image decoding method of claim 1,

wherein the slice mode information, the slice number information, and the tile index difference value information are signaled through a picture parameter set.

3. The image decoding method of claim 1,

wherein the method further comprises:

in response to the slice number information being greater than a pre-defined value, decoding tile index presence information indicating whether the tile index difference value information is present in the bitstream or not, and wherein the tile index difference value information is decoded from the bitstream only when the tile index presence information indicates that the tile index difference value information is present in the bitstream.

4. The image decoding method of claim 1,

wherein the index of the first tile in the second slice is derived by adding a value of the tile index difference value information to the index of the first tile in the first slice.

5. An image encoding method, the method comprising:

encoding slice mode information, the slice mode information representing which one of a rectangular slice mode and a raster scan order mode is applied to a current picture,

wherein under the rectangular slice mode, all of slices in the current picture are determined as a rectangular shape, and

wherein under the raster scan order mode, at least one slice in the current picture has a shape other than the rectangular shape;

in response to the slice mode information is encoded with a value indicating that the rectangular slice mode is applied, encoding slice number information, the slice number information representing a value subtracting 1 from a number of slices included in the current picture;

encoding tile index difference value information for a first slice in the current picture, the first slice being composed of one or more tile in the current picture,

wherein the tile index difference value information represents a difference between an index of a first tile in the first slice and an index of a first tile in a second slice in the current picture,

wherein the index of the first tile in the second slice is the same as a value adding the difference to the index of the first tile in the first slice, and

wherein the tile index difference value information is encoded into a bitstream in response to the number of slices in the current picture being greater than a pre-defined value.

43

6. The image encoding method of claim 5,  
 wherein the slice mode information, the slice number  
 information, and the tile index difference value infor-  
 mation are signaled through a picture parameter set.
7. The image encoding method of claim 5,  
 wherein the method further comprises:  
 in response to the number of slices in the current picture  
 being greater than a pre-defined value, encoding tile  
 index presence information indicating whether the tile  
 index difference value information is present in the  
 bitstream or not, and  
 wherein the tile index difference value information is  
 encoded into the bitstream only when the tile index  
 presence information is encoded with a value indicating  
 that the tile index difference value information is pres-  
 ent in the bitstream.
8. The image encoding method of claim 5,  
 wherein a value of the tile index difference value infor-  
 mation is determined by subtracting a value of the  
 index of the first tile in the first slice from the index of  
 the first tile in the second slice.
9. A non-transitory computer-readable recording medium,  
 comprising executable instructions that, when executed by a  
 processor, causes the processor to perform steps for storing  
 a bitstream generated by an image encoding method,  
 wherein the image encoding method comprises:  
 encoding slice mode information, the slice mode infor-  
 mation representing which one of a rectangular slice

44

- mode and a raster scan order mode is applied to a  
 current picture,  
 wherein under the rectangular slice mode, all of slices in  
 the current picture are determined as a rectangular  
 shape, and  
 wherein under the raster scan order mode, at least one  
 slice in the current picture has a shape other than the  
 rectangular shape;  
 in response to the slice mode information is encoded with  
 a value indicating that the rectangular slice mode is  
 applied, encoding slice number information, the slice  
 number information representing a value subtracting 1  
 from a number of slices included in the current picture;  
 encoding tile index difference value information for a first  
 slice in the current picture, the first slice being com-  
 posed of one or more tile in the current picture,  
 wherein the tile index difference value information rep-  
 resents a difference between an index of a first tile in  
 the first slice and an index of a first tile in a second slice  
 in the current picture,  
 wherein the index of the first tile in the second slice is the  
 same as a value adding the difference to the index of the  
 first tile in the first slice, and  
 wherein the tile index difference value information is  
 encoded into a bitstream in response to the number of  
 slices in the current picture being greater than a pre-  
 defined value.

\* \* \* \* \*