

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 991 337**

51 Int. Cl.:

H04N 19/96	(2014.01)
H04N 19/119	(2014.01)
H04N 19/174	(2014.01)
H04N 19/172	(2014.01)
H04N 19/44	(2014.01)
H04N 19/46	(2014.01)
H04N 19/176	(2014.01)
H04N 19/50	(2014.01)
H04N 19/132	(2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **20.09.2017** **E 21207296 (1)**

97 Fecha y número de publicación de la concesión europea: **18.09.2024** **EP 3975573**

54 Título: **Procedimiento de descodificación y procedimiento de codificación de una señal de vídeo**

30 Prioridad:

20.09.2016 KR 20160120080
20.09.2016 KR 20160120081

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
03.12.2024

73 Titular/es:

KT CORPORATION (100.0%)
90 Buljeong-roBundang-gu
Seongnam-si, Gyeonggi-do 13606, KR

72 Inventor/es:

LEE, BAE KEUN

74 Agente/Representante:

GONZÁLEZ PECES, Gustavo Adolfo

ES 2 991 337 T3

Aviso:En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín Europeo de Patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre Concesión de Patentes Europeas).

DESCRIPCIÓN

Procedimiento de descodificación y procedimiento de codificación de una señal de vídeo

Campo técnico

La presente invención se refiere a un procedimiento y un aparato para procesar una señal de vídeo.

5 Técnica antecedente

Recientemente, la demanda de imágenes de alta resolución y alta calidad, como imágenes de alta definición (HD) e imágenes de ultra alta definición (UHD), ha aumentado en diversos campos de aplicación. Sin embargo, los datos de imágenes de mayor resolución y calidad tienen cantidades de datos cada vez mayores en comparación con los datos de imágenes convencionales. Por lo tanto, cuando se transmiten datos de imagen utilizando un medio como las redes de banda ancha convencionales por cable e inalámbricas, o cuando se almacenan datos de imagen utilizando un medio de almacenamiento convencional, los costes de transmisión y almacenamiento aumentan. Para resolver estos problemas que se producen con el aumento de la resolución y la calidad de los datos de imagen, se pueden utilizar técnicas de codificación/decodificación de imágenes de alta eficacia.

La tecnología de compresión de imágenes incluye varias técnicas, entre ellas: una técnica de interpolación para predecir un valor de píxel incluido en una imagen actual a partir de una imagen anterior o posterior de la imagen actual; una técnica de intrapredicción para predecir un valor de píxel incluido en una imagen actual utilizando información de píxeles de la imagen actual; una técnica de codificación de entropía para asignar un código corto a un valor con una frecuencia de aparición alta y asignar un código largo a un valor con una frecuencia de aparición baja; etc. Los datos de imagen pueden comprimirse eficazmente utilizando dicha tecnología de compresión de imágenes, y pueden transmitirse o almacenarse.

Mientras tanto, con la demanda de imágenes de alta resolución, también ha aumentado la demanda de contenidos de imágenes estereográficas, que es un nuevo servicio de imágenes. Se discute una técnica de compresión de vídeo para proporcionar eficazmente contenidos de imagen estereográfica con alta resolución y ultra alta resolución.

El documento WO 2016/091161 A1 divulga un procedimiento de codificación de vídeo mediante un proceso de partición de bloques que incluye un proceso de partición de árbol binario. El proceso de partición de bloques se aplica a un bloque de datos de vídeo para particionar el bloque en sub-bloques finales. El proceso de codificación que comprende el proceso de predicción, el proceso de transformación o ambos para el bloque se aplicará en el nivel final de los sub-bloques. El proceso de partición en árbol binario puede aplicarse a un bloque dado de forma recursiva para generar nodos de hoja de árbol binario hasta que se cumpla una condición de terminación.

30 Divulgación

Problema técnico

Un objeto de la presente invención es proporcionar un procedimiento y un aparato para dividir eficientemente un bloque objetivo de codificación/decodificación en la codificación/decodificación de una señal de vídeo.

Un objeto de la presente invención es proporcionar un procedimiento y un aparato para dividir un bloque objetivo de codificación/decodificación en dos bloques de tipo simétrico o de tipo asimétrico en la codificación/decodificación de una señal de vídeo.

Los objetos técnicos que se pretenden alcanzar con la presente invención no se limitan a los problemas técnicos mencionados. Y, otros problemas técnicos que no se mencionan serán evidentemente comprendidos por los expertos en la materia a partir de la siguiente descripción.

40 Solución técnica

La invención se define mediante reivindicaciones independientes y varios aspectos de la invención se divulgan en las reivindicaciones dependientes, la descripción y los dibujos.

Efectos ventajosos

Según la presente invención, la eficiencia de codificación/decodificación puede mejorarse dividiendo eficientemente un bloque objetivo de codificación/decodificación.

Según la presente invención, la eficiencia de codificación/decodificación puede mejorarse dividiendo un bloque objetivo de codificación/decodificación en un tipo simétrico o un tipo asimétrico.

Los efectos obtenibles por la presente invención no se limitan a los efectos arriba mencionados, y otros efectos no mencionados pueden ser claramente comprendidos por los expertos en la materia a partir de la descripción que sigue.

Descripción de los dibujos

- 5 La FIG. 1 es un diagrama de bloques que ilustra un dispositivo para codificar un vídeo según una realización de la presente invención.
- La FIG. 2 es un diagrama de bloques que ilustra un dispositivo para decodificar un vídeo según una realización de la presente invención.
- La FIG. 3 es un diagrama que ilustra un ejemplo de partición jerárquica de un bloque de codificación basado en una estructura de árbol según una realización de la presente invención.
- La FIG. 4 es un diagrama que ilustra un tipo de partición en el que se permite la partición basada en árbol binario según una realización de la presente invención.
- 10 La FIG. 5 es un diagrama que ilustra un ejemplo en el que sólo se permite una partición basada en un árbol binario de un tipo predeterminado según una realización de la presente invención.
- La FIG. 6 es un diagrama para explicar un ejemplo en el que se codifica/decodifica información relacionada con el número permitido de particiones de árbol binario, según una realización a la que se aplica la presente invención.
- 15 La FIG. 7 ilustra un tipo de partición de un bloque de codificación basado en la partición de árbol binario asimétrico.
- La FIG. 8 muestra un ejemplo en el que un bloque de codificación se divide en una pluralidad de bloques de codificación utilizando QTBT y partición de árbol binario asimétrico.
- La FIG. 9 es un diagrama que ilustra los tipos de partición que pueden aplicarse a un bloque de codificación.
- 20 La FIG. 10 es un diagrama que ilustra un modo de partición que puede aplicarse a un bloque de codificación cuando el bloque de codificación se codifica mediante interpredicción.
- La FIG. 11 es un diagrama de flujo que ilustra los procesos de partición de un bloque de codificación según una realización de la presente invención.
- 25 La FIG. 12 es un diagrama de flujo que ilustra los procesos de obtención de una muestra residual según una realización a la que se aplica la presente invención.

Modo de la invención

- Una variedad de modificaciones puede ser hecha a la presente invención y hay varias realizaciones de la presente invención, los ejemplos de los cuales ahora serán proporcionados con referencia a dibujos y descritos en detalle.
- 30 Los términos utilizados en la presente memoria descriptiva, "primero", "segundo", etc. pueden utilizarse para describir varios componentes, pero los componentes no deben interpretarse como limitados a los términos. Los términos sólo se utilizan para diferenciar un componente de otros. Por ejemplo, el "primer" componente puede denominarse "segundo" componente sin apartarse del ámbito de la presente invención, y el "segundo" componente también puede denominarse de forma similar "primer" componente. El término "y/o" incluye una combinación de una pluralidad de elementos o cualquiera de una pluralidad de términos.
- 35 Se entenderá que cuando en la presente descripción se hace referencia a un elemento simplemente como "conectado a" o "acoplado a" otro elemento sin estar "directamente conectado a" o "directamente acoplado a" otro elemento, puede estar "directamente conectado a" o "directamente acoplado a" otro elemento o estar conectado a o acoplado a otro elemento, interponiéndose el otro elemento entre ambos. Por el contrario, debe entenderse que cuando se hace referencia a un elemento como "directamente acoplado" o "directamente conectado" a otro elemento, no hay elementos intermedios presentes.
- 40 Los términos utilizados en la presente memoria descriptiva se utilizan meramente para describir realizaciones particulares, y no pretenden limitar la presente invención. Una expresión utilizada en singular engloba la expresión del plural, a menos que tenga un significado claramente diferente en el contexto. En la presente memoria, debe entenderse que términos como "incluyendo", "teniendo", etc. tienen por objeto indicar la existencia de las características, números, etapas, acciones, elementos, partes o combinaciones de los mismos que se describen en la memoria, y no pretenden excluir la posibilidad de que puedan existir o añadirse una o más características, números, etapas, acciones, elementos, partes o combinaciones de los mismos.
- 45 A continuación, se describirán en detalle realizaciones preferentes de la presente invención con referencia a los dibujos adjuntos. En lo sucesivo, los mismos elementos constitutivos en los dibujos se denotan por los mismos números de referencia, y se omitirá una descripción repetida de los mismos elementos.
- 50 La FIG. 1 es un diagrama de bloques que ilustra un dispositivo para codificar un vídeo según una realización de la presente invención.
- En referencia a la FIG. 1, el dispositivo 100 para codificar un vídeo puede incluir: un módulo de partición de imágenes 110, módulos de predicción 120 y 125, un módulo de transformación 130, un módulo de cuantización 135, un módulo de reordenación 160, un módulo de codificación de entropía 165, un módulo de cuantización inversa 140, un módulo de transformación inversa 145, un módulo de filtrado 150 y una memoria 155.
- 55

Las partes constituyentes mostradas en la FIG. 1 se muestran de forma independiente para representar funciones características diferentes entre sí en el dispositivo para codificar un vídeo. Por lo tanto, no significa que cada parte constituyente esté constituida en una unidad constituyente de hardware o software separado. En otras palabras, cada parte constituyente incluye cada una de las partes constituyentes enumeradas por conveniencia. Así, al menos dos partes constituyentes de cada parte constituyente pueden combinarse para formar una parte constituyente o una parte constituyente puede dividirse en una pluralidad de partes constituyentes para realizar cada función. La realización en la que cada parte constituyente se combina y la realización en la que una parte constituyente se divide también se incluyen en el ámbito de la presente invención, siempre que no se aparten de la esencia de la presente invención.

Asimismo, algunos de los constituyentes pueden no ser constituyentes indispensables que realicen funciones esenciales de la presente invención, sino constituyentes selectivos que mejoren únicamente el rendimiento de la misma.

El módulo de partición de imágenes 110 puede particionar una imagen de entrada en una o más unidades de procesamiento. Aquí, la unidad de procesamiento puede ser una unidad de predicción (PU), una unidad de transformación (TU) o una unidad de codificación (CU). El módulo de partición de imágenes 110 puede particionar una imagen en combinaciones de múltiples unidades de codificación, unidades de predicción y unidades de transformación, y puede codificar una imagen seleccionando una combinación de unidades de codificación, unidades de predicción y unidades de transformación con un criterio predeterminado (por ejemplo, función de coste).

Por ejemplo, una imagen puede dividirse en múltiples unidades de codificación. Se puede utilizar una estructura de árbol recursiva, como una estructura de árbol cuádruple, para dividir una imagen en unidades de codificación. Una unidad de codificación que se particiona en otras unidades de codificación con una imagen o una unidad de codificación mayor como raíz puede particionarse con nodos hijos correspondientes al número de unidades de codificación particionadas. Una unidad de codificación que ya no está particionada por una limitación predeterminada sirve como nodo hoja. Es decir, cuando se supone que sólo es posible la partición cuadrada para una unidad de codificación, una unidad de codificación puede particionarse en otras cuatro unidades de codificación como máximo.

En lo sucesivo, en la realización de la presente invención, la unidad de codificación puede significar una unidad que realiza la codificación, o una unidad que realiza la decodificación.

Una unidad de predicción puede ser una de las particiones divididas en un cuadrado o una forma rectangular con el mismo tamaño en una única unidad de codificación, o una unidad de predicción puede ser una de las particiones divididas para tener una forma / tamaño diferente en una única unidad de codificación.

Cuando una unidad de predicción sometida a intrapredicción se genera basándose en una unidad de codificación y la unidad de codificación no es la unidad de codificación más pequeña, la intrapredicción puede realizarse sin particionar la unidad de codificación en múltiples unidades de predicción NxN.

Los módulos de predicción 120 y 125 pueden incluir un módulo de predicción intermedia 120 que realiza la predicción intermedia y un módulo de predicción interna 125 que realiza la predicción interna. Se puede determinar si se va a realizar una predicción inter o intra para la unidad de predicción, y se puede determinar información detallada (por ejemplo, un modo de predicción intra, un vector de movimiento, una imagen de referencia, etc.) de acuerdo con cada procedimiento de predicción. Aquí, la unidad de procesamiento sometida a predicción puede ser diferente de la unidad de procesamiento para la que se determina el procedimiento de predicción y el contenido detallado. Por ejemplo, el procedimiento de predicción, el modo de predicción, etc. pueden ser determinados por la unidad de predicción, y la predicción puede ser realizada por la unidad de transformación. Se puede introducir en el módulo de transformación 130 un valor residual (bloque residual) entre el bloque de predicción generado y un bloque original. Además, el módulo de codificación de entropía 165 puede codificar la información del modo de predicción, la información del vector de movimiento, etc. utilizada para la predicción con el valor residual y transmitirla a un dispositivo para decodificar un vídeo. Cuando se utiliza un modo de codificación particular, es posible transmitir a un dispositivo para decodificar vídeo codificando el bloque original tal cual sin generar el bloque de predicción a través de los módulos de predicción 120 y 125.

El módulo de interpredicción 120 puede predecir la unidad de predicción basándose en información de al menos una de las imágenes anteriores o posteriores de la imagen actual, o puede predecir la unidad de predicción basándose en información de algunas regiones codificadas de la imagen actual, en algunos casos. El módulo de interpredicción 120 puede incluir un módulo de interpolación de imagen de referencia, un módulo de predicción de movimiento y un módulo de compensación de movimiento.

El módulo de interpolación de imagen de referencia puede recibir información de imagen de referencia de la memoria 155 y puede generar información de píxel de un píxel entero o menor que el píxel entero de la imagen de referencia. En el caso de los píxeles luma, se puede utilizar un filtro de interpolación basado en DCT de 8 tomas con diferentes coeficientes de filtro para generar información de píxeles de un píxel entero o inferior a un píxel entero en unidades de 1/4 de píxel. En el caso de las señales de croma, se puede utilizar un filtro de interpolación basado en DCT de 4 tomas con diferentes coeficientes de filtro para generar información de píxeles de un píxel entero o inferior a un píxel entero en unidades de 1/8 de píxel.

5 El módulo de predicción de movimiento puede realizar la predicción de movimiento basándose en la imagen de referencia interpolada por el módulo de interpolación de imagen de referencia. Como procedimientos para calcular un vector de movimiento, pueden utilizarse diversos procedimientos, como un algoritmo de correspondencia de bloques basado en la búsqueda completa (FBMA), una búsqueda en tres pasos (TSS), un nuevo algoritmo de búsqueda en tres pasos (NTS), etc. El vector de movimiento puede tener un valor de vector de movimiento en unidades de 1/2 píxel o 1/4 de píxel basado en un píxel interpolado. El módulo de predicción de movimiento puede predecir una unidad de predicción actual cambiando el procedimiento de predicción de movimiento. Como procedimientos de predicción de movimiento, se pueden utilizar varios procedimientos, como un procedimiento de salto, un procedimiento de fusión, un procedimiento AMVP (Advanced Motion Vector Prediction), un procedimiento de copia intrabloque, etc.

10 El módulo de predicción intra 125 puede generar una unidad de predicción basada en información de píxel de referencia vecina a un bloque actual que es información de píxel en la imagen actual. Cuando el bloque vecino de la unidad de predicción actual es un bloque sometido a interpredicción y, por tanto, un píxel de referencia es un píxel sometido a interpredicción, el píxel de referencia incluido en el bloque sometido a interpredicción puede sustituirse por información de píxel de referencia de un bloque vecino sometido a intrapredicción. Es decir, cuando un píxel de referencia no está disponible, se puede utilizar al menos un píxel de referencia de los píxeles de referencia disponibles en lugar de la información del píxel de referencia no disponible.

15 Los modos de predicción en la predicción intra pueden incluir un modo de predicción direccional que utiliza información de píxel de referencia dependiendo de una dirección de predicción y un modo de predicción no direccional que no utiliza información direccional al realizar la predicción. Un modo para predecir la información de luma puede ser diferente de un modo para predecir la información de croma, y para predecir la información de croma, se puede utilizar la información del modo de predicción intra utilizado para predecir la información de luma o la información de la señal de luma predicha.

20 Al realizar la intrapredicción, cuando el tamaño de la unidad de predicción es el mismo que el tamaño de la unidad de transformación, la intrapredicción puede realizarse en la unidad de predicción basándose en los píxeles situados a la izquierda, arriba a la izquierda y arriba de la unidad de predicción. Sin embargo, al realizar la intrapredicción, cuando el tamaño de la unidad de predicción es diferente del tamaño de la unidad de transformación, la intrapredicción puede realizarse utilizando un píxel de referencia basado en la unidad de transformación. Además, la intrapredicción mediante partición NxN puede utilizarse sólo para la unidad de codificación más pequeña.

25 En el procedimiento de predicción intra, se puede generar un bloque de predicción después de aplicar un filtro AIS (Adaptive Intra Smoothing) a un píxel de referencia dependiendo de los modos de predicción. El tipo de filtro AIS aplicado al píxel de referencia puede variar. Para llevar a cabo el procedimiento de predicción intra, se puede predecir un modo de predicción intra de la unidad de predicción actual a partir del modo de predicción intra de la unidad de predicción vecina a la unidad de predicción actual. En la predicción del modo de predicción de la unidad de predicción actual utilizando la información de modo predicha desde la unidad de predicción vecina, cuando el modo de predicción intra de la unidad de predicción actual es el mismo que el modo de predicción intra de la unidad de predicción vecina, la información que indica que los modos de predicción de la unidad de predicción actual y de la unidad de predicción vecina son iguales entre sí puede transmitirse utilizando información de bandera predeterminada. Cuando el modo de predicción de la unidad de predicción actual es diferente del modo de predicción de la unidad de predicción vecina, se puede realizar la codificación de entropía para codificar la información del modo de predicción del bloque actual.

30 Además, un bloque residual que incluye información sobre un valor residual que es diferente entre la unidad de predicción sometida a predicción y el bloque original de la unidad de predicción puede generarse basándose en las unidades de predicción generadas por los módulos de predicción 120 y 125. El bloque residual generado puede introducirse en el módulo de transformación 130.

35 El módulo de transformación 130 puede transformar el bloque residual que incluye la información sobre el valor residual entre el bloque original y la unidad de predicción generada por los módulos de predicción 120 y 125 utilizando un procedimiento de transformación, como la transformada discreta del coseno (DCT), la transformada discreta del seno (DST) y la KLT. La aplicación de DCT, DST o KLT para transformar el bloque residual puede determinarse basándose en la información del modo de predicción intra de la unidad de predicción utilizada para generar el bloque residual.

40 El módulo de cuantización 135 puede cuantizar valores transformados a un dominio de frecuencia por el módulo de transformación 130. Los coeficientes de cuantización pueden variar según el bloque o la importancia de una imagen. Los valores calculados por el módulo de cuantización 135 pueden proporcionarse al módulo de cuantización inversa 140 y al módulo de reordenación 160.

El módulo de reordenación 160 puede reordenar los coeficientes de los valores residuales cuantizados.

45 El módulo de reordenación 160 puede cambiar un coeficiente en forma de bloque bidimensional a un coeficiente en forma de vector unidimensional mediante un procedimiento de exploración de coeficientes. Por ejemplo, el módulo de reordenamiento 160 puede escanear desde un coeficiente de CC a un coeficiente en un dominio de alta frecuencia utilizando un procedimiento de escaneo en zigzag para cambiar los coeficientes a la forma de vectores unidimensionales. Dependiendo del tamaño de la unidad de transformación y del modo de predicción intra, en lugar

de la exploración en zigzag puede utilizarse la exploración en dirección vertical, en la que los coeficientes en forma de bloques bidimensionales se exploran en la dirección de la columna, o la exploración en dirección horizontal, en la que los coeficientes en forma de bloques bidimensionales se exploran en la dirección de la fila. Es decir, se puede determinar qué procedimiento de escaneo se utiliza entre el escaneo en zigzag, el escaneo en dirección vertical y el escaneo en dirección horizontal, dependiendo del tamaño de la unidad de transformación y del modo de predicción intra.

El módulo de codificación de entropía 165 puede realizar la codificación de entropía basándose en los valores calculados por el módulo de reordenación 160. La codificación de entropía puede utilizar varios procedimientos de codificación, por ejemplo, la codificación exponencial de Golomb, la codificación de longitud variable adaptable al contexto (CAVLC) y la codificación aritmética binaria adaptable al contexto (CABAC).

El módulo de codificación de entropía 165 puede codificar una variedad de información, como información de coeficiente de valor residual e información de tipo de bloque de la unidad de codificación, información de modo de predicción, información de unidad de partición, información de unidad de predicción, información de unidad de transformación, información de vector de movimiento, información de marco de referencia, información de interpolación de bloque, información de filtrado, etc. del módulo de reordenación 160 y los módulos de predicción 120 y 125.

El módulo de codificación de entropía 165 puede codificar con entropía los coeficientes de la entrada de la unidad de codificación desde el módulo de reordenación 160.

El módulo de cuantización inversa 140 puede cuantizar inversamente los valores cuantizados por el módulo de cuantización 135 y el módulo de transformación inversa 145 puede transformar inversamente los valores transformados por el módulo de transformación 130. El valor residual generado por el módulo de cuantización inversa 140 y el módulo de transformación inversa 145 puede combinarse con la unidad de predicción predicha por un módulo de estimación de movimiento, un módulo de compensación de movimiento, y el módulo de intrapredicción de los módulos de predicción 120 y 125 de forma que pueda generarse un bloque reconstruido.

El módulo de filtrado 150 puede incluir al menos uno de un filtro de desbloqueo, una unidad de corrección de desplazamiento y un filtro de bucle adaptativo (ALF).

El filtro de desbloqueo puede eliminar la distorsión de bloque que se produce debido a los límites entre los bloques en la imagen reconstruida. Para determinar si se debe realizar el desbloqueo, los píxeles incluidos en varias filas o columnas del bloque pueden ser una base para determinar si se debe aplicar el filtro de desbloqueo al bloque actual. Cuando se aplica el filtro de desbloqueo al bloque, puede aplicarse un filtro fuerte o débil en función de la fuerza de filtrado de desbloqueo requerida. Además, al aplicar el filtro de desbloqueo, el filtrado de dirección horizontal y el filtrado de dirección vertical pueden procesarse en paralelo.

El módulo de corrección de desplazamiento puede corregir el desplazamiento con la imagen original en unidades de un píxel en la imagen sometida a desbloqueo. Para realizar la corrección del desplazamiento en una imagen concreta, es posible utilizar un procedimiento de aplicación del desplazamiento teniendo en cuenta la información de los bordes de cada píxel o un procedimiento de división de los píxeles de una imagen en un número predeterminado de regiones, determinando una región que debe someterse al desplazamiento y aplicando el desplazamiento a la región determinada.

El filtrado de bucle adaptativo (ALF) puede realizarse basándose en el valor obtenido al comparar la imagen reconstruida filtrada y la imagen original. Los píxeles incluidos en la imagen pueden dividirse en grupos predeterminados, puede determinarse un filtro que se aplicará a cada uno de los grupos y el filtrado puede realizarse individualmente para cada grupo. Las unidades de codificación (CU) pueden transmitir información sobre si aplicar ALF y una señal de luma. La forma y el coeficiente de filtrado de un filtro para ALF pueden variar en función de cada bloque. Además, el filtro para ALF de la misma forma (forma fija) puede aplicarse independientemente de las características del bloque de destino de la aplicación.

La memoria 155 puede almacenar el bloque reconstruido o la imagen calculada mediante el módulo de filtrado 150. El bloque o imagen reconstruida almacenada puede proporcionarse a los módulos de predicción 120 y 125 para realizar la interpredicción.

La FIG. 2 es un diagrama de bloques que ilustra un dispositivo para descodificar un vídeo según una realización de la presente invención.

En referencia a la FIG. 2, el dispositivo 200 para descodificar un vídeo puede incluir: un módulo de descodificación de entropía 210, un módulo de reordenación 215, un módulo de cuantización inversa 220, un módulo de transformación inversa 225, módulos de predicción 230 y 235, un módulo de filtrado 240, y una memoria 245.

Cuando se introduce un flujo de bits de vídeo desde el dispositivo para codificar un vídeo, el flujo de bits de entrada puede descodificarse según un proceso inverso del dispositivo para codificar un vídeo.

5 El módulo de decodificación de entropía 210 puede realizar la decodificación de entropía según un proceso inverso de la codificación de entropía por el módulo de codificación de entropía del dispositivo para codificar un vídeo. Por ejemplo, en correspondencia con los procedimientos realizados por el dispositivo para codificar un vídeo, pueden aplicarse diversos procedimientos, como la codificación exponencial de Golomb, la codificación de longitud variable adaptable al contexto (CAVLC) y la codificación aritmética binaria adaptable al contexto (CABAC).

El módulo de decodificación de entropía 210 puede decodificar información sobre la predicción intra e inter realizada por el dispositivo para codificar un vídeo.

10 El módulo de reordenación 215 puede realizar una reordenación en la entropía del flujo de bits decodificada por el módulo de decodificación de entropía 210 basándose en el procedimiento de reordenación utilizado en el dispositivo para codificar un vídeo. El módulo de reordenación puede reconstruir y reordenar los coeficientes en forma de vectores unidimensionales al coeficiente en forma de bloques bidimensionales. El módulo de reordenación 215 puede recibir información relacionada con el escaneo de coeficientes realizado en el dispositivo para codificar un vídeo y puede realizar la reordenación mediante un procedimiento de escaneo inverso de los coeficientes basado en el orden de escaneo realizado en el dispositivo para codificar un vídeo.

15 El módulo de cuantización inversa 220 puede realizar una cuantización inversa basada en un parámetro de cuantización recibido del dispositivo para codificar un vídeo y los coeficientes reordenados del bloque.

20 El módulo de transformación inversa 225 puede realizar la transformación inversa, es decir, DCT inversa, DST inversa y KLT inversa, que es el proceso inverso de transformación, es decir, DCT, DST y KLT, realizado por el módulo de transformación sobre el resultado de cuantización por el dispositivo para codificar un vídeo. La transformación inversa puede realizarse basándose en una unidad de transferencia determinada por el dispositivo para codificar un vídeo. El módulo de transformación inversa 225 del dispositivo para decodificar un vídeo puede realizar selectivamente esquemas de transformación (por ejemplo, DCT, DST y KLT) dependiendo de múltiples informaciones, como el procedimiento de predicción, el tamaño del bloque actual, la dirección de predicción, etc.

25 Los módulos de predicción 230 y 235 pueden generar un bloque de predicción basado en la información sobre la generación de bloques de predicción recibida del módulo de descodificación de entropía 210 y la información de bloque o imagen descodificada previamente recibida de la memoria 245.

30 Como se ha descrito anteriormente, al igual que el funcionamiento del dispositivo para codificar un vídeo, al realizar la predicción intra, cuando el tamaño de la unidad de predicción es el mismo que el tamaño de la unidad de transformación, la predicción intra puede realizarse en la unidad de predicción basándose en los píxeles situados a la izquierda, arriba a la izquierda y arriba de la unidad de predicción. Al realizar la intrapredicción, cuando el tamaño de la unidad de predicción es diferente del tamaño de la unidad de transformación, la intrapredicción puede realizarse utilizando un píxel de referencia basado en la unidad de transformación. Además, la intrapredicción mediante partición NxN puede utilizarse sólo para la unidad de codificación más pequeña.

35 Los módulos de predicción 230 y 235 pueden incluir un módulo de determinación de la unidad de predicción, un módulo de predicción inter, y un módulo de predicción intra. El módulo de determinación de la unidad de predicción puede recibir una variedad de información, como información sobre la unidad de predicción, información sobre el modo de predicción de un procedimiento de predicción intra, información sobre la predicción de movimiento de un procedimiento de predicción inter, etc. del módulo de descodificación de entropía 210, puede dividir una unidad de codificación actual en unidades de predicción, y puede determinar si se realiza la predicción inter o la predicción intra en la unidad de predicción. Utilizando la información requerida en la interpredicción de la unidad de predicción actual recibida del dispositivo para codificar un vídeo, el módulo de interpredicción 230 puede realizar la interpredicción en la unidad de predicción actual basándose en la información de al menos una de una imagen anterior o una imagen posterior de la imagen actual que incluye la unidad de predicción actual. Alternativamente, la interpredicción puede realizarse basándose en la información de algunas regiones prerreconstruidas en la imagen actual incluyendo la unidad de predicción actual.

Para realizar la interpredicción, puede determinarse para la unidad de codificación cuál de los modos de salto, fusión, AMVP y copia entre bloques se utiliza como procedimiento de predicción de movimiento de la unidad de predicción incluida en la unidad de codificación.

50 El módulo de predicción intra 235 puede generar un bloque de predicción basado en la información de píxeles de la imagen actual. Cuando la unidad de predicción es una unidad de predicción sometida a intrapredicción, la intrapredicción puede realizarse basándose en la información del modo de intrapredicción de la unidad de predicción recibida del dispositivo para codificar un vídeo. El módulo de predicción intra 235 puede incluir un filtro de suavizado adaptativo intra (AIS), un módulo de interpolación de píxeles de referencia y un filtro DC. El filtro AIS realiza el filtrado en el píxel de referencia del bloque actual, y la aplicación del filtro puede determinarse en función del modo de predicción de la unidad de predicción actual. El filtrado AIS puede realizarse en el píxel de referencia del bloque actual utilizando el modo de predicción de la unidad de predicción y la información del filtro AIS recibida del dispositivo para codificar un vídeo. Cuando el modo de predicción del bloque actual es un modo en el que no se realiza el filtrado AIS, puede que no se aplique el filtro AIS.

- 5 Cuando el modo de predicción de la unidad de predicción es un modo de predicción en el que la predicción intra se realiza basándose en el valor de píxel obtenido interpolando el píxel de referencia, el módulo de interpolación de píxel de referencia puede interpolar el píxel de referencia para generar el píxel de referencia de un píxel entero o menor que un píxel entero. Cuando el modo de predicción de la unidad de predicción actual es un modo de predicción en el que se genera un bloque de predicción sin interpolar el píxel de referencia, el píxel de referencia puede no interpolarse. El filtro de CC puede generar un bloque de predicción mediante filtrado cuando el modo de predicción del bloque actual es un modo de CC.
- 10 El bloque o imagen reconstruida puede proporcionarse al módulo de filtrado 240. El módulo de filtrado 240 puede incluir el filtro de desbloqueo, el módulo de corrección de desplazamiento y el ALF.
- 15 La información sobre si el filtro de desbloqueo se aplica o no al bloque o imagen correspondiente y la información sobre cuál de un filtro fuerte y cuál de un filtro débil se aplica cuando se aplica el filtro de desbloqueo puede recibirse del dispositivo para codificar un vídeo. El filtro de desbloqueo del dispositivo para decodificar un vídeo puede recibir información sobre el filtro de desbloqueo del dispositivo para codificar un vídeo, y puede realizar el filtrado de desbloqueo en el bloque correspondiente.
- 15 El módulo de corrección de desplazamiento puede realizar la corrección de desplazamiento en la imagen reconstruida basándose en el tipo de corrección de desplazamiento y en la información de valor de desplazamiento aplicada a una imagen al realizar la codificación.
- 20 El ALF puede aplicarse a la unidad de codificación basándose en la información sobre si aplicar el ALF, la información sobre el coeficiente ALF, etc. recibida del dispositivo para codificar un vídeo. La información ALF puede proporcionarse como incluida en un conjunto de parámetros concreto.
- La memoria 245 puede almacenar la imagen o bloque reconstruido para su uso como imagen o bloque de referencia, y puede proporcionar la imagen reconstruida a un módulo de salida.
- 25 Como se ha descrito anteriormente, en la realización de la presente invención, por conveniencia de explicación, la unidad de codificación se utiliza como un término que representa una unidad para la codificación, pero la unidad de codificación puede servir como una unidad que realiza la decodificación, así como la codificación.
- Además, un bloque actual puede representar un bloque objetivo a codificar/decodificar. Y, el bloque actual puede representar un bloque de árbol de codificación (o una unidad de árbol de codificación), un bloque de codificación (o una unidad de codificación), un bloque de transformación (o una unidad de transformación), un bloque de predicción (o una unidad de predicción), o similares dependiendo de una etapa de codificación/decodificación.
- 30 Una imagen puede codificarse/decodificarse dividiéndose en bloques de base con forma cuadrada o no cuadrada. En este momento, el bloque base puede denominarse unidad de árbol de codificación. La unidad de árbol de codificación puede definirse como una unidad de codificación del mayor tamaño permitido dentro de una secuencia o una rebanada. La información relativa a si la unidad de árbol de codificación tiene una forma cuadrada o no cuadrada o la información relativa a un tamaño de la unidad de árbol de codificación puede señalarse a través de un conjunto de parámetros de
- 35 secuencia, un conjunto de parámetros de imagen o una cabecera de corte. La unidad de árbol de codificación puede dividirse en particiones de menor tamaño. En este momento, si se supone que la profundidad de una partición generada al dividir la unidad de árbol de codificación es 1, la profundidad de una partición generada al dividir la partición que tiene profundidad 1 puede definirse como 2. Es decir, una partición generada al dividir una partición que tiene una profundidad k en la unidad de árbol de codificación puede definirse como que tiene una profundidad $k+1$.
- 40 Una partición de tamaño arbitrario generada al dividir una unidad de árbol de codificación puede definirse como una unidad de codificación. La unidad de codificación puede dividirse recursivamente o dividirse en unidades base para realizar predicción, cuantización, transformación o filtrado en bucle, y similares. Por ejemplo, una partición de tamaño arbitrario generada al dividir la unidad de codificación puede definirse como una unidad de codificación, o puede definirse como una unidad de transformación o una unidad de predicción, que es una unidad base para realizar
- 45 predicción, cuantización, transformación o filtrado en bucle y similares.
- La partición de una unidad de árbol de codificación o de una unidad de codificación puede realizarse basándose en al menos una de una línea vertical y una línea horizontal. Además, el número de líneas verticales u horizontales que dividen la unidad de árbol de codificación o la unidad de codificación puede ser al menos uno o más. Por ejemplo, la unidad de árbol de codificación o la unidad de codificación puede dividirse en dos particiones utilizando una línea
- 50 vertical o una línea horizontal, o la unidad de árbol de codificación o la unidad de codificación puede dividirse en tres particiones utilizando dos líneas verticales o dos líneas horizontales. Alternativamente, la unidad de árbol de codificación o la unidad de codificación puede dividirse en cuatro particiones que tengan una longitud y una anchura de $1/2$ utilizando una línea vertical y una línea horizontal.
- 55 Cuando una unidad de árbol de codificación o una unidad de codificación se divide en una pluralidad de particiones utilizando al menos una línea vertical o al menos una línea horizontal, las particiones pueden tener un tamaño uniforme o un tamaño diferente. Alternativamente, cualquier partición puede tener un tamaño diferente de las particiones restantes.

En las realizaciones descritas a continuación, se supone que una unidad de árbol de codificación o una unidad de codificación se divide en una estructura de árbol cuádruple o una estructura de árbol binaria. Sin embargo, también es posible dividir una unidad de árbol de codificación o una unidad de codificación utilizando un mayor número de líneas verticales o un mayor número de líneas horizontales.

- 5 La FIG. 3 es un diagrama que ilustra un ejemplo de partición jerárquica de un bloque de codificación basado en una estructura de árbol según una realización de la presente invención.

Una señal de vídeo de entrada se decodifica en unidades de bloque predeterminadas. Dicha unidad predeterminada para descodificar la señal de vídeo de entrada es un bloque de codificación. El bloque de codificación puede ser una unidad que realice predicción intra/inter, transformación y cuantización. Además, se determina un modo de predicción (por ejemplo, modo de predicción intra o modo de predicción inter) en unidades de un bloque de codificación, y los bloques de predicción incluidos en el bloque de codificación pueden compartir el modo de predicción determinado. El bloque de codificación puede ser un bloque cuadrado o no cuadrado con un tamaño arbitrario en un rango de 8×8 a 64×64 , o puede ser un bloque cuadrado o no cuadrado con un tamaño de 128×128 , 256×256 , o más.

- 10 Específicamente, el bloque de codificación puede ser particionado jerárquicamente basándose en al menos uno de un árbol cuádruple y un árbol binario. Aquí, la partición basada en árbol cuádruple puede significar que un bloque de codificación $2N \times 2N$ se particiona en cuatro bloques de codificación $N \times N$, y la partición basada en árbol binario puede significar que un bloque de codificación se particiona en dos bloques de codificación. Aunque se realice la partición basada en el árbol binario, puede existir un bloque de codificación con forma cuadrada en la profundidad inferior.

- 15 La partición basada en árboles binarios puede realizarse simétrica o asimétricamente. Además, el bloque de codificación particionado basado en el árbol binario puede ser un bloque cuadrado o un bloque no cuadrado, como una forma rectangular. Por ejemplo, un tipo de partición en el que se permite la partición basada en árbol binario puede comprender al menos una de las siguientes: tipo simétrico de $2N \times N$ (unidad de codificación no cuadrada de dirección horizontal) o $N \times 2N$ (unidad de codificación no cuadrada de dirección vertical), tipo asimétrico de $nL \times 2N$, $nR \times 2N$, $2N \times nU$, o $2N \times nD$.

- 20 La partición basada en árboles binarios puede limitarse a una partición de tipo simétrico o asimétrico. En este caso, la construcción de la unidad de árbol de codificación con bloques cuadrados puede corresponder a la partición CU de árbol cuádruple, y la construcción de la unidad de árbol de codificación con bloques simétricos no cuadrados puede corresponder a la partición de árbol binario. La construcción de la unidad de árbol de codificación con bloques cuadrados y bloques simétricos no cuadrados puede corresponder a la partición de CU de árbol cuádruple y binario.

- 25 El particionamiento basado en árboles binarios puede realizarse en un bloque de codificación en el que ya no se realiza el particionamiento basado en árboles cuádruples. La partición basada en el árbol cuádruple ya no puede realizarse en el bloque de codificación particionado en base al árbol binario.

- Además, la partición de una profundidad inferior puede determinarse en función de un tipo de partición de una profundidad superior. Por ejemplo, si se permite la partición basada en árboles binarios en dos o más profundidades, sólo podrá permitirse en la profundidad inferior el mismo tipo que la partición basada en árboles binarios de la profundidad superior. Por ejemplo, si la partición basada en árboles binarios en la profundidad superior se realiza con el tipo $2N \times N$, la partición basada en árboles binarios en la profundidad inferior también se realiza con el tipo $2N \times N$. Alternativamente, si la partición basada en árboles binarios en la profundidad superior se realiza con el tipo $N \times 2N$, la partición basada en árboles binarios en la profundidad inferior también se realiza con el tipo $N \times 2N$.

- 30 Por el contrario, también es posible permitir, en una profundidad inferior, sólo un tipo diferente de un tipo de partición de árbol binario de una profundidad superior.

- Puede ser posible limitar sólo un tipo específico de partición basada en árbol binario a utilizar para secuencia, rebanada, unidad de árbol de codificación o unidad de codificación. A modo de ejemplo, para la unidad de árbol de codificación sólo puede permitirse el tipo $2N \times N$ o el tipo $N \times 2N$ de partición basada en árbol binario. Un tipo de partición disponible puede estar predefinido en un codificador o un decodificador. O la información sobre el tipo de partición disponible o sobre el tipo de partición no disponible se puede codificar y luego señalar a través de un flujo de bits.

- 35 La FIG. 5 es un diagrama que ilustra un ejemplo en el que sólo se permite un tipo específico de partición basada en árboles binarios. La FIG. 5A muestra un ejemplo en el que sólo se permite el tipo $N \times 2N$ de partición basada en árbol binario, y la FIG. 5B muestra un ejemplo en el que sólo se permite la partición basada en árboles binarios de tipo $2N \times N$. Para implementar la partición adaptativa basada en el árbol cuádruple o en el árbol binario, puede utilizarse información que indique la partición basada en el árbol cuádruple, información sobre el tamaño/profundidad del bloque de codificación en el que se permite la partición basada en el árbol cuádruple, información que indique la partición basada en el árbol binario, información sobre el tamaño/profundidad del bloque de codificación en el que se permite la partición basada en el árbol binario, información sobre el tamaño/profundidad del bloque de codificación en el que no se permite la partición basada en el árbol binario, información sobre si la partición basada en el árbol binario se realiza en dirección vertical, en dirección horizontal, o similares. Por ejemplo, `quad_split_flag` indica si el bloque de codificación está dividido en cuatro bloques de codificación, y `binary_split_flag` indica si el bloque de codificación está

dividido en dos bloques de codificación. Cuando el bloque de codificación se divide en dos bloques de codificación, puede señalarse `is_hor_split_flag`, que indica si una dirección de partición del bloque de codificación es una dirección vertical o una dirección horizontal.

5 Además, se puede obtener información sobre el número de veces que se permite una partición de árbol binario, una profundidad a la que se permite la partición de árbol binario, o el número de las profundidades a las que se permite la partición de árbol binario para una unidad de árbol de codificación o una unidad de codificación específica. La información puede codificarse en unidades de una unidad de árbol de codificación o en una unidad de codificación, y puede transmitirse a un descodificador a través de un flujo de bits.

10 Por ejemplo, una sintaxis "`max_binary_depth_idx_minus1`" que indica una profundidad máxima a la que se permite la partición en árbol binario puede codificarse / descodificarse a través de un flujo de bits. En este caso, `max_binary_depth_idx_minus1 + 1` puede indicar la profundidad máxima a la que se permite la partición del árbol binario.

15 Refiriéndose al ejemplo mostrado en la FIG. 6, en la FIG. 6, la partición del árbol binario se ha realizado para una unidad de codificación que tiene una profundidad de 2 y una unidad de codificación que tiene una profundidad de 3. En consecuencia, al menos una de las informaciones que indican el número de veces que se ha realizado la partición en árbol binario en la unidad de árbol de codificación (es decir, 2 veces), la información que indica la profundidad máxima a la que se ha permitido la partición en árbol binario en la unidad de árbol de codificación (es decir, profundidad 3), o el número de profundidades a las que se ha realizado la partición en árbol binario en la unidad de árbol de codificación (es decir, 2 (profundidad 2 y profundidad 3)) puede codificarse / descodificarse a través de un flujo de bits.

20 Como otro ejemplo, al menos una de las informaciones sobre el número de veces que se permite la partición en árbol binario, la profundidad a la que se permite la partición en árbol binario, o el número de las profundidades a las que se permite la partición en árbol binario puede obtenerse para cada secuencia o cada rebanada. Por ejemplo, la información puede codificarse en unidades de una secuencia, una imagen o una unidad de corte y transmitirse a través de un flujo de bits. En consecuencia, al menos uno de los números de la partición de árbol binario en una primera rebanada, la profundidad máxima en la que se permite la partición de árbol binario en la primera rebanada, o el número de profundidades en las que se realiza la partición de árbol binario en la primera rebanada puede ser diferencia de una segunda rebanada. Por ejemplo, en la primera porción, la partición de árbol binario puede permitirse sólo para una profundidad, mientras que en la segunda porción, la partición de árbol binario puede permitirse para dos profundidades.

30 Como otro ejemplo, el número de veces que se permite la partición en árbol binario, la profundidad a la que se permite la partición en árbol binario, o el número de profundidades a las que se permite la partición en árbol binario pueden establecerse de forma diferente según un identificador de nivel temporal (`TemporalID`) de una rebanada o una imagen. Aquí, el identificador de nivel temporal (`TemporalID`) se utiliza para identificar cada una de una pluralidad de capas de vídeo que tienen una escalabilidad de al menos una de vista, espacial, temporal o de calidad.

35 Como se muestra en la FIG. 3, el primer bloque de codificación 300 con la profundidad de partición (profundidad de división) de k puede particionarse en múltiples segundos bloques de codificación basados en el árbol cuádruple. Por ejemplo, los segundos bloques de codificación 310 a 340 pueden ser bloques cuadrados que tengan la mitad de anchura y la mitad de altura del primer bloque de codificación, y la profundidad de partición del segundo bloque de codificación puede aumentarse a $k+1$.

40 El segundo bloque de codificación 310 con la profundidad de partición de $k+1$ puede ser particionado en múltiples terceros bloques de codificación con la profundidad de partición de $k+2$. El particionamiento del segundo bloque de codificación 310 puede realizarse utilizando selectivamente uno de los árboles cuádruples y el árbol binario en función de un procedimiento de particionamiento. Aquí, el procedimiento de partición puede determinarse basándose en al menos una de la información que indica la partición basada en árbol cuádruple y la información que indica la partición basada en árbol binario.

45 Cuando el segundo bloque de codificación 310 es particionado basado en el árbol cuádruple, el segundo bloque de codificación 310 puede ser particionado en cuatro terceros bloques de codificación 310a que tienen la mitad del ancho y la mitad de la altura del segundo bloque de codificación, y la profundidad de partición del tercer bloque de codificación 310a puede ser incrementada a $k+2$. Por el contrario, cuando el segundo bloque de codificación 310 se particiona basándose en el árbol binario, el segundo bloque de codificación 310 puede particionarse en dos terceros bloques de codificación. Aquí, cada uno de los dos terceros bloques de codificación puede ser un bloque no cuadrado que tenga una de las mitades de anchura y de altura del segundo bloque de codificación, y la profundidad de partición puede aumentarse a $k+2$. El segundo bloque de codificación puede determinarse como un bloque no cuadrado de una dirección horizontal o de una dirección vertical dependiendo de una dirección de partición, y la dirección de partición puede determinarse basándose en la información sobre si la partición basada en árboles binarios se realiza en una dirección vertical o en una dirección horizontal.

Mientras tanto, el segundo bloque de codificación 310 puede determinarse como un bloque de codificación de hoja que ya no está particionado basado en el árbol cuádruple o el árbol binario. En este caso, el bloque de codificación de hojas puede utilizarse como bloque de predicción o como bloque de transformación.

5 Al igual que la partición del segundo bloque de codificación 310, el tercer bloque de codificación 310a puede determinarse como un bloque de codificación de hoja, o puede particionarse aún más basándose en el árbol cuádruple o el árbol binario.

10 Mientras tanto, el tercer bloque de codificación 310b particionado basado en el árbol binario puede ser particionado adicionalmente en bloques de codificación 310b-2 de una dirección vertical o bloques de codificación 310b-3 de una dirección horizontal basado en el árbol binario, y la profundidad de partición de los bloques de codificación relevantes puede ser incrementada a $k+3$. Alternativamente, el tercer bloque de codificación 310b puede determinarse como un bloque de codificación de hoja 310b-1 que ya no está particionado basado en el árbol binario. En este caso, el bloque de codificación 310b-1 puede utilizarse como un bloque de predicción o un bloque de transformación. Sin embargo, el proceso de partición anterior puede realizarse de forma limitada basándose en al menos una de las informaciones sobre el tamaño/profundidad del bloque de codificación que permite la partición cuádruple en árbol, la información sobre el tamaño/profundidad del bloque de codificación que permite la partición binaria en árbol y la información sobre el tamaño/profundidad del bloque de codificación que no permite la partición binaria en árbol.

15 Un número de un candidato que representa un tamaño de un bloque de codificación puede estar limitado a un número predeterminado, o un tamaño de un bloque de codificación en una unidad predeterminada puede tener un valor fijo. Por ejemplo, el tamaño del bloque de codificación en una secuencia o en una imagen puede limitarse a 256x256, 128x128 o 32x32. La información que indica el tamaño del bloque de codificación en la secuencia o en la imagen puede señalarse mediante una cabecera de secuencia o una cabecera de imagen.

20 Como resultado de la partición basada en un árbol cuádruple y un árbol binario, una unidad de codificación puede representarse como una forma cuadrada o rectangular de tamaño arbitrario.

25 Como resultado de una división basada en el árbol cuádruple y el árbol binario, un bloque de codificación que no se divide más puede utilizarse como un bloque de predicción o un bloque de transformación. Es decir, en un procedimiento de partición QTBT basado en un árbol cuádruple y un árbol binario, un bloque de codificación puede convertirse en un bloque de predicción y un bloque de predicción puede convertirse en un bloque de transformación. Por ejemplo, cuando se utiliza el procedimiento de partición QTBT, puede generarse una imagen de predicción en una unidad de un bloque de codificación, y una señal residual, que es una diferencia entre una imagen original y la imagen de predicción, se transforma en una unidad de un bloque de codificación. Aquí, generar la imagen de predicción en una unidad de un bloque de codificación puede significar que se determina la información de movimiento para un bloque de codificación o que se determina un modo de predicción intra para un bloque de codificación.

30 En el procedimiento de partición QTBT, puede establecerse que sólo se permita la partición simétrica en BT. Sin embargo, si sólo se permite la partición binaria simétrica aunque un objeto y un fondo se dividan en un límite de bloque, la eficacia de la codificación puede disminuir. Por consiguiente, en la presente invención, se propone un procedimiento de partición asimétrica de un bloque de codificación para aumentar la eficacia de la codificación.

35 La partición de árbol binario asimétrico representa la división de un bloque de codificación en dos bloques de codificación más pequeños. Como resultado de la partición del árbol binario asimétrico, el bloque de codificación puede dividirse en dos bloques de codificación de forma asimétrica. Por comodidad de explicación, en las siguientes realizaciones, dividir un bloque de codificación en dos particiones de forma simétrica se denominará partición de árbol binario (o partición de árbol binario), y dividir un bloque de codificación en dos particiones de forma asimétrica se denominará partición de árbol binario asimétrico (o partición de árbol binario asimétrico).

40 La FIG. 7 ilustra un tipo de partición de un bloque de codificación basado en la partición de árbol binario asimétrico. Un bloque de codificación de $2N \times 2N$ puede dividirse en dos bloques de codificación cuya relación de anchura sea $n:(1-n)$ o dos bloques de codificación cuya relación de altura es $n:(1-n)$. Donde n puede representar un número real mayor que 0 y menor que 1.

45 En la FIG. 7 que dos bloques de codificación cuya relación de anchura es 1:3 o 3:1 o cuya relación de altura es 1:3 o 3:1 se generan aplicando la partición de árbol binario asimétrico a un bloque de codificación.

50 Específicamente, como un bloque de codificación de tamaño $W \times H$ es particionado en una dirección vertical, una partición izquierda cuyo ancho es $1/4W$ y una partición derecha cuyo ancho es $3/4W$ pueden ser generadas. Como se ha descrito anteriormente, un tipo de partición en el que la anchura de la partición izquierda es menor que la anchura de la partición derecha puede denominarse partición binaria $nL \times 2N$.

55 Cuando un bloque de codificación de tamaño $W \times H$ se particiona en dirección vertical, puede generarse una partición izquierda cuya anchura es $3/4W$ y una partición derecha cuya anchura es $1/4W$. Como se ha descrito anteriormente, un tipo de partición en el que la anchura de la partición derecha es menor que la anchura de la partición izquierda puede denominarse partición binaria $nR \times 2N$.

Como un bloque de codificación de tamaño $W \times H$ es particionado en una dirección horizontal, una partición superior cuyo ancho es $1/4H$ y una partición inferior cuyo ancho es $3/4H$ pueden ser generadas. Como se ha descrito anteriormente, un tipo de partición en el que la altura de la partición superior es menor que la altura de la partición inferior puede denominarse partición binaria $2N \times nU$.

5 Como un bloque de codificación de tamaño $W \times H$ se particiona en una dirección horizontal, se puede generar una partición superior cuya anchura es $3/4H$ y una partición inferior cuya anchura es $1/4H$. Como se ha descrito anteriormente, un tipo de partición en el que la altura de la partición inferior es menor que la altura de la partición superior puede denominarse partición binaria $2N \times nD$.

10 En la FIG. 7, se ilustra que una relación de anchura o una relación de altura entre dos bloques de codificación es 1:3 o 3:1. Sin embargo, la relación de anchura o la relación de altura entre dos bloques de codificación generados por partición de árbol binario asimétrico no está limitada a ello. El bloque de codificación puede dividirse en dos bloques de codificación con una relación de anchura o altura diferente a las mostradas en la FIG. 7.

15 Cuando se utiliza la partición binaria asimétrica en árbol, puede determinarse un tipo de partición binaria asimétrica de un bloque de codificación basándose en la información señalada a través de un flujo de bits. Por ejemplo, un tipo de partición de un bloque de codificación puede determinarse basándose en información que indique una dirección de partición del bloque de codificación e información que indique si una primera partición, generada al dividir el bloque de codificación, tiene un tamaño menor que una segunda partición.

20 La información que indica la dirección de partición del bloque de codificación puede ser una bandera de 1 bit que indica si el bloque de codificación está particionado en dirección vertical o en dirección horizontal. Por ejemplo, `hor_binary_flag` puede indicar si el bloque de codificación está particionado en dirección horizontal. Si un valor de `hor_binary_flag` es 1, puede indicar que el bloque de codificación está particionado en la dirección horizontal y si el valor de `hor_binary_flag` es 0, puede indicar que el bloque de codificación está particionado en la dirección vertical. Como alternativa, puede utilizarse `ver_binary_flag`, que indica si el bloque de codificación está dividido o no en la dirección vertical.

25 La información que indica si la primera partición tiene un tamaño menor que la segunda partición puede ser una bandera de 1 bit. Por ejemplo, `is_left_above_small_part_flag` puede indicar si el tamaño de una partición izquierda o superior generada al dividir el bloque de codificación es menor que una partición derecha o inferior. Si un valor de `is_left_above_small_part_flag` es 1, significa que el tamaño de la partición izquierda o superior es menor que el de la partición derecha o inferior. Si el valor de `is_left_above_small_part_flag` es 0, significa que el tamaño de la partición izquierda o superior es mayor que el de la partición derecha o inferior. Como alternativa, se puede utilizar `is_right_bottom_small_part_flag` que indica si el tamaño de la partición derecha o inferior es menor que el de la partición izquierda o superior.

30 Alternativamente, los tamaños de una primera partición y una segunda partición pueden determinarse utilizando información que indique una relación de anchura, una relación de altura o una relación de área entre la primera partición y la segunda partición.

35 Cuando un valor de `hor_binary_flag` es 0 y un valor de `is_left_above_small_part_flag` es 1, puede representar $nL \times 2N$ partición binaria, y cuando un valor de `hor_binary_flag` es 0 y un valor de `is_left_above_small_part_flag` es 0, puede representar $nR \times 2N$ partición binaria. Además, cuando un valor de `hor_binary_flag` es 1 y un valor de `is_left_above_small_part_flag` es 1, puede representar una partición binaria $2N \times nU$, y cuando un valor de `hor_binary_flag` es 1 y un valor de `is_left_above_small_part_flag` es 0, puede representar una partición binaria $2N \times nD$.

40 Como otro ejemplo, el tipo de partición binaria asimétrica del bloque de codificación puede determinarse mediante información de índice que indica un tipo de partición del bloque de codificación. Aquí, la información de índice es información que debe señalarse a través de un flujo de bits, y puede codificarse con una longitud fija (es decir, un número fijo de bits) o puede codificarse con una longitud variable. Por ejemplo, la Tabla 1 muestra el índice de partición para cada partición binaria asimétrica.

[Tabla 1]

	Índice de partición asimétrica	Binarización
$nL \times 2N$	0	0
$nR \times 2N$	1	10
$2N \times nU$	2	100
$2N \times nD$	3	111

5 Se puede utilizar la partición de árbol binario asimétrico dependiendo del procedimiento de partición QTBT. Por ejemplo, si la partición en árbol binario o en árbol cuádruple ya no se aplica al bloque de codificación, puede determinarse si se aplica o no la partición en árbol binario asimétrico al bloque de codificación. En este caso, la aplicación o no de la partición de árbol binario asimétrico al bloque de codificación puede determinarse mediante información señalada a través del flujo de bits. Por ejemplo, la información puede ser una bandera de 1 bit 'asymmetric_binary_tree_flag', y basándose en la bandera, se puede determinar si la partición de árbol binario asimétrico se aplica al bloque de codificación.

10 Alternativamente, cuando se determina que el bloque de codificación está particionado en dos bloques, se puede determinar si el tipo de partición es partición de árbol binario o partición de árbol binario asimétrico. Aquí, si el tipo de partición del bloque de codificación es la partición de árbol binario o la partición de árbol binario asimétrico puede determinarse mediante información señalada a través del flujo de bits. Por ejemplo, la información puede ser una bandera de 1 bit 'is_asymmetric_split_flag', y en base a la bandera, se puede determinar si el bloque de codificación se divide en una forma simétrica o asimétrica de.

15 Como otro ejemplo, los índices asignados a particiones binarias de tipo simétrico y a particiones binarias de tipo asimétrico pueden ser diferentes, y puede determinarse basándose en la información de índice si el bloque de codificación está particionado en un tipo simétrico o en un tipo asimétrico. Por ejemplo, la Tabla 2 muestra un ejemplo en el que se asignan diferentes índices a particiones de tipo binario simétrico y particiones de tipo binario asimétrico.

[Tabla 2]

	Índice de partición binaria	Binarización
2NxN (partición binaria en dirección horizontal)	0	0
Nx2N (partición binaria en dirección vertical)	1	10
nLx2N	2	110
nRx2N	3	1110
2NxnU	4	11110
2NxnD	5	11111

20 Un bloque de árbol de codificación o un bloque de codificación puede dividirse en una pluralidad de bloques de codificación mediante partición en árbol cuádruple, partición en árbol binario o partición en árbol binario asimétrico. Por ejemplo, FIG. 8 muestra un ejemplo en el que un bloque de codificación se divide en una pluralidad de bloques de codificación utilizando QTBT y partición de árbol binario asimétrico. En referencia a la FIG. 8, se puede observar que la partición de árbol binario asimétrico se realiza en partición de profundidad 2 en el primer dibujo, partición de profundidad 3 en el segundo dibujo, y partición de profundidad 3 en el tercer dibujo, respectivamente.

25 Puede restringirse que un bloque de codificación dividido por la partición de árbol binario asimétrico deje de estar dividido. Por ejemplo, la información relacionada con un árbol cuaternario, árbol binario, o árbol binario asimétrico puede no ser codificada/decodificada para un bloque de codificación que es generado por la partición de árbol binario asimétrico. Es decir, para un bloque de codificación generado a través de la partición de árbol binario asimétrico, puede omitirse una bandera que indique si se aplica la partición de árbol cuádruple, una bandera que indique si se aplica la partición de árbol binario, una bandera que indique si se aplica la partición de árbol binario asimétrico, una bandera que indique una dirección de la partición de árbol binario o de la partición de árbol binario asimétrico, o información de índice que indique una partición binaria asimétrica, o similares.

30 Como otro ejemplo, el permitir o no la partición del árbol binario puede determinarse dependiendo de si el QTBT está permitido o no. Por ejemplo, en una imagen o corte en el que no se utilice el procedimiento de partición basado en QTBT, puede estar restringido no utilizar la partición de árbol binario asimétrico.

35 La información que indica si se permite la partición asimétrica en árbol binario puede codificarse y señalizarse en una unidad de un bloque, una rebanada o una imagen. Aquí, la información que indica si se permite la partición asimétrica del árbol binario puede ser una bandera de 1 bit. Por ejemplo, si un valor de is_used_asymmetric_QTBT_enabled_flag es 0, puede indicar que no se utiliza la partición de árbol binario asimétrico. También es posible que is_used_asymmetric_QTBT_enabled_Flag se ponga a 0 sin señalización de ello cuando no se utiliza la partición en árbol binario en una imagen o en una rebanada.

También es posible determinar un tipo de partición permitido en un bloque de codificación basado en un tamaño, una forma, una profundidad de partición, o un tipo de partición del bloque de codificación. Por ejemplo, al menos uno de los tipos de partición, formas de partición o número de particiones permitidas en un bloque de codificación generado por la partición en árbol cuádruple y en un bloque de codificación generado por la partición en árbol binario pueden ser diferentes entre sí.

Por ejemplo, si un bloque de codificación es generado por la partición de árbol cuádruple, toda la partición de árbol cuádruple, la partición de árbol binario, y la partición de árbol binario asimétrico pueden ser permitidas para el bloque de codificación. Es decir, si se genera un bloque de codificación basado en la partición de árbol cuádruple, todos los tipos de partición mostrados en la FIG. 9 puede aplicarse al bloque de codificación. Por ejemplo, una partición $2N \times 2N$ puede representar un caso en el que un bloque de codificación no se divide más, $N \times N$ puede representar un caso en el que un bloque de codificación se particiona en un árbol cuádruple, y $N \times 2N$ y $2N \times N$ pueden representar un caso en el que un bloque de codificación se particiona en un árbol binario. Además, $nL \times 2N$, $nR \times 2N$, $2N \times nU$ y $2N \times nD$ pueden representar casos en los que un bloque de codificación está particionado en un árbol binario asimétrico.

Por otro lado, cuando un bloque de codificación es generado por la partición de árbol binario, puede no estar permitido utilizar la partición de árbol binario asimétrico para el bloque de codificación. Es decir, cuando el bloque de codificación se genera basándose en la partición de árbol binario, puede restringirse para no aplicar el tipo de partición asimétrica ($nL \times 2N$, $nR \times 2N$, $2N \times nU$, $2N \times nD$) entre los tipos de partición mostrados en la FIG. 9 al bloque de codificación.

Cuando se utiliza QTBT, un bloque de codificación que no se divide más puede utilizarse como bloque de predicción. Es decir, el bloque de codificación puede codificarse utilizando al menos uno de los siguientes procedimientos: un modo de omisión, una predicción intra, una predicción inter o un procedimiento de omisión.

Como otro ejemplo, si se determina un bloque de codificación, se puede determinar un bloque de predicción que tenga el mismo tamaño que el bloque de codificación o un tamaño menor que el bloque de codificación a través de la partición predictiva del bloque de codificación. La partición predictiva de un bloque de codificación puede realizarse mediante un modo de partición (Part_mode) que indica un tipo de partición del bloque de codificación. El tamaño o la forma de un bloque de predicción puede determinarse en función del modo de partición del bloque de codificación. El tipo de partición del bloque de codificación puede determinarse mediante información que especifique cualquiera de los candidatos a partición. En este momento, los candidatos a partición disponibles para el bloque de codificación pueden incluir un tipo de partición asimétrica (por ejemplo, $nL \times 2N$, $nR \times 2N$, $2N \times nU$, $2N \times nD$) dependiendo de un tamaño, una forma o un modo de codificación del bloque de codificación. Por ejemplo, las particiones candidatas disponibles para un bloque de codificación pueden determinarse en función de un modo de codificación de un bloque actual. Por ejemplo, FIG. 10 es un diagrama que ilustra un modo de partición que puede aplicarse a un bloque de codificación cuando el bloque de codificación se codifica mediante interpredicción.

Si un bloque de codificación es codificado por una interpredicción, uno de los 8 modos de partición ilustrados en la FIG. 10 puede aplicarse al bloque de codificación.

Por otra parte, cuando un bloque de codificación se codifica por intrapredicción, puede aplicarse al bloque de codificación un modo de partición de PART_2Nx2N o PART_NxN.

El modo PART_NxN puede aplicarse cuando un bloque de codificación tiene un tamaño mínimo. En este caso, el tamaño mínimo del bloque de codificación puede estar predefinido en el codificador y el decodificador. Alternativamente, la información relativa al tamaño mínimo del bloque de codificación puede señalarse a través del flujo de bits. Por ejemplo, el tamaño mínimo del bloque de codificación se señala a través de una cabecera de rebanada, de modo que el tamaño mínimo del bloque de codificación puede definirse por rebanada.

En otro ejemplo, las particiones candidatas disponibles para un bloque de codificación pueden determinarse de forma diferente dependiendo de al menos uno de los tamaños o formas del bloque de codificación. Por ejemplo, el número o tipo de particiones candidatas disponibles para un bloque de codificación puede determinarse de forma diferente en función de al menos uno de los factores siguientes: tamaño o forma del bloque de codificación.

Alternativamente, un tipo o número de candidatos de partición asimétricos entre los candidatos de partición disponibles para un bloque de codificación puede ser limitado dependiendo de un tamaño o forma del bloque de codificación. Por ejemplo, el número o tipo de candidatos a partición asimétrica disponibles para un bloque de codificación puede determinarse de forma diferente en función de, al menos, uno de los factores siguientes: el tamaño o la forma del bloque de codificación.

En general, un bloque de predicción puede tener un tamaño de 64×64 a 4×4 . Sin embargo, cuando un bloque de codificación se codifica mediante interpredicción, es posible evitar que el bloque de predicción tenga un tamaño de 4×4 para reducir el ancho de banda de la memoria al realizar la compensación de movimiento.

La FIG. 11 es un diagrama de flujo que ilustra los procesos de partición de un bloque de codificación según una realización de la presente invención.

En primer lugar, puede determinarse si se realiza la partición en árbol cuádruple en un bloque actual S1110. Si se determina que la partición en árbol cuádruple se va a realizar en el bloque actual, el bloque actual se puede dividir en cuatro bloques de codificación S1120.

5 Por otro lado, si se determina que la partición en árbol cuádruple no está permitida para el bloque actual, puede determinarse si se realiza partición en árbol binario o partición en árbol binario asimétrico en el bloque actual S1130.

10 Si se determina que la partición binaria en árbol o la partición binaria asimétrica en árbol se realiza para el bloque actual, puede determinarse un tipo de partición binaria del bloque actual S1140. En este momento, el tipo de partición de árbol binario del bloque actual puede determinarse basándose en al menos una de las informaciones que indican información de partición del bloque actual, información que indica un tamaño de partición dividido, o información de índice que especifica un tipo de partición.

El bloque actual puede dividirse en dos bloques simétricos o asimétricos según el tipo de partición binaria S1150 determinado.

15 Es posible determinar secuencialmente si se realiza o no la partición en árbol binario para el bloque actual y si se realiza o no la partición en árbol binario asimétrico para el bloque actual. Por ejemplo, se puede determinar si realizar o no la partición de árbol binario asimétrico sólo cuando se determina que la partición de árbol binario no está permitida para el bloque actual.

La FIG. 12 es un diagrama de flujo que ilustra los procesos de obtención de una muestra residual según una realización a la que se aplica la presente invención.

20 En primer lugar, puede obtenerse un coeficiente residual de un bloque actual S1210. Un decodificador puede obtener un coeficiente residual mediante un procedimiento de exploración de coeficientes. Por ejemplo, el decodificador puede realizar un escaneo de coeficientes utilizando un escaneo diagonal, un escaneo jig-zag, un escaneo arriba-derecha, un escaneo vertical o un escaneo horizontal, y puede obtener coeficientes residuales en forma de bloque bidimensional.

Se puede realizar una cuantización inversa en el coeficiente residual del bloque actual S1220.

25 Es posible determinar si omitir una transformada inversa sobre el coeficiente residual descuantizado del bloque actual S1230. En concreto, el decodificador puede determinar si omitir la transformación inversa en al menos una de las direcciones horizontal o vertical del bloque actual. Cuando se determina aplicar la transformada inversa en al menos una de las direcciones horizontal o vertical del bloque actual, se puede obtener una muestra residual del bloque actual transformando inversamente el coeficiente residual decuantizado del bloque actual S1240. Aquí, la transformada inversa se puede realizar utilizando al menos una de DCT, DST y KLT.

30 Cuando la transformación inversa se omite tanto en la dirección horizontal como en la dirección vertical del bloque actual, la transformación inversa no se realiza en la dirección horizontal ni en la dirección vertical del bloque actual. En este caso, la muestra residual del bloque actual puede obtenerse escalando el coeficiente residual descuantizado con un valor predeterminado S1250.

35 Omitir la transformación inversa en la dirección horizontal significa que la transformación inversa no se realiza en la dirección horizontal pero sí en la dirección vertical. En este momento, se puede realizar el escalado en dirección horizontal.

40 Omitir la transformación inversa en la dirección vertical significa que la transformación inversa no se realiza en la dirección vertical pero sí en la dirección horizontal. En este momento, se puede realizar el escalado en dirección vertical.

45 Puede determinarse si puede utilizarse o no una técnica de salto por transformada inversa para el bloque actual dependiendo de un tipo de partición del bloque actual. Por ejemplo, si el bloque actual se genera mediante una partición basada en un árbol binario, el esquema de salto de la transformada inversa puede estar restringido para el bloque actual. En consecuencia, cuando el bloque actual se genera mediante la partición basada en el árbol binario, la muestra residual del bloque actual puede obtenerse mediante la transformación inversa del bloque actual. Además, cuando el bloque actual se genera mediante partición basada en árbol binario, puede omitirse la codificación/decodificación de la información que indica si se omite o no la transformación inversa (por ejemplo, transform_skip_flag).

50 Alternativamente, cuando el bloque actual se genera mediante partición basada en árbol binario, es posible limitar el esquema de salto de transformada inversa a al menos una de las direcciones horizontal o vertical. Aquí, la dirección en la que se limita el esquema de salto de transformada inversa puede determinarse basándose en información decodificada del flujo de bits, o puede determinarse adaptativamente basándose en al menos uno de un tamaño del bloque actual, una forma del bloque actual, o un modo de intrapredicción del bloque actual.

Por ejemplo, cuando el bloque actual es un bloque no cuadrado que tiene una anchura mayor que una altura, el esquema de salto de transformación inversa puede permitirse sólo en la dirección vertical y restringirse en la dirección horizontal. Es decir, cuando el bloque actual es $2N \times N$, la transformación inversa se realiza en la dirección horizontal del bloque actual, y la transformación inversa puede realizarse selectivamente en la dirección vertical.

5 Por otro lado, cuando el bloque actual es un bloque no cuadrado que tiene una altura mayor que una anchura, el esquema de salto de transformación inversa puede permitirse sólo en la dirección horizontal y restringirse en la dirección vertical. Es decir, cuando el bloque actual es $N \times 2N$, la transformación inversa se realiza en la dirección vertical del bloque actual, y la transformación inversa puede realizarse selectivamente en la dirección horizontal.

10 En contraste con el ejemplo anterior, cuando el bloque actual es un bloque no cuadrado que tiene una anchura mayor que una altura, el esquema de salto de transformación inversa puede permitirse sólo en la dirección horizontal, y cuando el bloque actual es un bloque no cuadrado que tiene una altura mayor que una anchura, el esquema de salto de transformación inversa puede permitirse sólo en la dirección vertical.

15 La información que indica si omitir o no la transformación inversa con respecto a la dirección horizontal o la información que indica si omitir la transformación inversa con respecto a la dirección vertical puede señalarse a través de un flujo de bits. Por ejemplo, la información que indica si omitir o no la transformación inversa en la dirección horizontal es un indicador de 1 bit, 'hor_transform_skip_flag', y la información que indica si omitir la transformación inversa en la dirección vertical es un indicador de 1 bit, 'ver_transform_skip_flag'. El codificador puede codificar al menos una de las opciones "hor_transform_skip_flag" o "ver_transform_skip_flag" en función de la forma del bloque actual. Además, el descodificador puede determinar si se omite o no la transformación inversa en la dirección horizontal o en la dirección vertical utilizando al menos uno de los indicadores "hor_transform_skip_flag" o "ver_transform_skip_flag".

20 Puede configurarse para omitir la transformada inversa para cualquier dirección del bloque actual dependiendo de un tipo de partición del bloque actual. Por ejemplo, si el bloque actual se genera mediante una partición basada en un árbol binario, puede omitirse la transformación inversa en la dirección horizontal o en la dirección vertical. Es decir, si el bloque actual se genera mediante partición basada en árbol binario, puede determinarse que la transformada inversa del bloque actual se omite en al menos una de las direcciones horizontal o vertical sin codificar/decodificar información (por ejemplo, transform_skip_flag, hor_transform_skip_flag, ver_transform_skip_flag) que indique si se omite o no la transformada inversa del bloque actual.

25 Aunque las realizaciones descritas anteriormente se han descrito sobre la base de una serie de etapas o diagramas de flujo, no limitan el orden de las series temporales de la invención, y pueden realizarse simultáneamente o en diferentes órdenes según sea necesario. Además, cada uno de los componentes (por ejemplo, unidades, módulos, etc.) que constituyen el diagrama de bloques en las realizaciones descritas anteriormente puede ser implementado por un dispositivo de hardware o software, y una pluralidad de componentes. También pueden combinarse varios componentes y aplicarse mediante un único dispositivo de hardware o software. Las realizaciones descritas anteriormente pueden implementarse en forma de instrucciones de programa que pueden ejecutarse a través de diversos componentes informáticos y grabarse en un medio de grabación legible por ordenador. El medio de grabación legible por ordenador puede incluir uno o una combinación de comandos de programa, archivos de datos, estructuras de datos y similares. Ejemplos de medios legibles por ordenador incluyen medios magnéticos como discos duros, disquetes y cintas magnéticas, medios de grabación óptica como CD-ROM y DVD, medios magneto-ópticos como discos flopticos, soportes y dispositivos de hardware específicamente configurados para almacenar y ejecutar instrucciones de programa como ROM, RAM, memoria flash y similares. El dispositivo de hardware puede configurarse para funcionar como uno o más módulos de software para realizar el procedimiento según la presente invención, y viceversa.

Aplicabilidad industrial

La presente invención puede aplicarse a dispositivos electrónicos capaces de codificar/decodificar un vídeo.

REIVINDICACIONES

1. Un procedimiento para decodificar un vídeo, comprendiendo el procedimiento:

decodificar una primera bandera, que indica si un bloque actual está dividido en dos particiones o no, a partir de un flujo de bits;

5 determinar, basándose en la primera bandera, si se divide o no el bloque actual en las dos particiones, **caracterizado por**

determinar, basándose en una segunda bandera señalada a través del flujo de bits, si el bloque actual se divide simétrica o asimétricamente cuando se determina dividir el bloque actual en las dos particiones;

10 determinar un tipo de partición asimétrica del bloque actual basándose en una tercera bandera y una cuarta bandera señaladas a través del flujo de bits cuando se determina que el bloque actual está dividido asimétricamente, utilizándose la tercera bandera para determinar si el bloque actual está dividido en dirección vertical o en dirección horizontal y utilizándose la cuarta bandera para determinar las ubicaciones de las dos particiones; y

15 dividir el bloque actual en una primera partición que tenga un tamaño de 1/4 del bloque actual y una segunda partición que tenga un tamaño de 3/4 del bloque actual,

en el que cuando el bloque actual se divide en la dirección horizontal, se determina, basándose en la cuarta bandera, si la primera partición o la segunda partición está situada en un lado superior o en un lado inferior, y en el que cuando el bloque actual se divide en la dirección vertical, se determina, basándose en la cuarta bandera, si la primera partición o la segunda partición está situada en un lado izquierdo o en un lado derecho.

20 2. El procedimiento de la reivindicación 1, en el que un número de tipos de partición asimétrica disponibles para el bloque actual se determina en función de un tamaño del bloque actual.

3. Un procedimiento para codificar un vídeo, comprendiendo el procedimiento:

codificar, en un flujo de bits, una primera bandera, que indica si un bloque actual está dividido en dos particiones o no,

25 **caracterizado por**

codificar una segunda bandera, que especifica si el bloque actual se divide simétrica o asimétricamente, en el flujo de bits cuando el bloque actual se divide en las dos particiones;

codificar en el flujo de bits, una tercera bandera que indica si el bloque actual está dividido en dirección vertical o en dirección horizontal; y

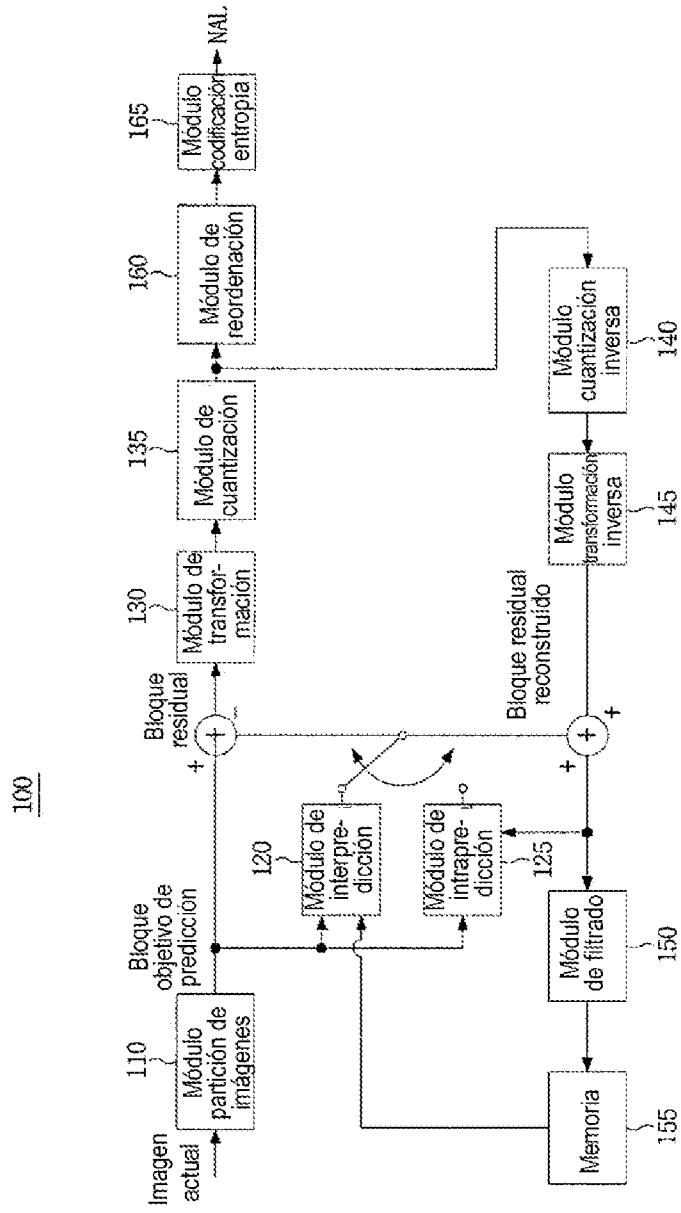
30 codificar en el flujo de bits, una cuarta bandera, que especifica las ubicaciones de las dos particiones, en el que cuando el bloque actual se divide asimétricamente, las dos particiones incluyen una primera partición que tiene un tamaño de 1/4 del bloque actual y una segunda partición que tiene un tamaño de 3/4 del bloque actual,

35 en el que cuando el bloque actual se divide en la dirección horizontal, la cuarta bandera especifica si la primera partición o la segunda partición se encuentra en un lado superior o en un lado inferior, y en el que cuando el bloque actual se divide en la dirección vertical, la cuarta bandera especifica si la primera partición o la segunda partición se encuentra en un lado izquierdo o en un lado derecho.

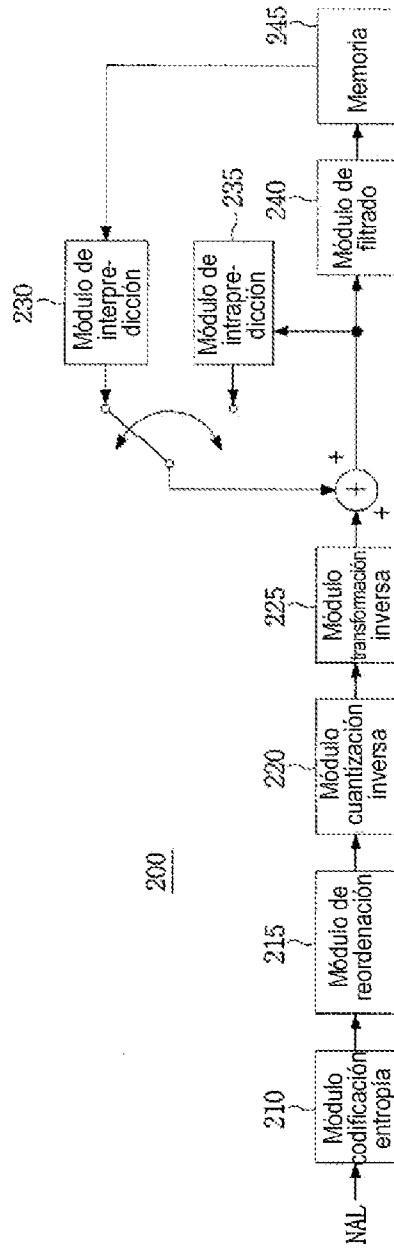
4. El procedimiento de la reivindicación 3, en el que un número de tipos de partición asimétrica disponibles para el bloque actual se determina en función de un tamaño del bloque actual.

40 5. Un medio legible por ordenador que comprende instrucciones que, cuando son ejecutadas por un ordenador, hacen que el ordenador lleve a cabo el procedimiento según cualquiera de las reivindicaciones 1-4.

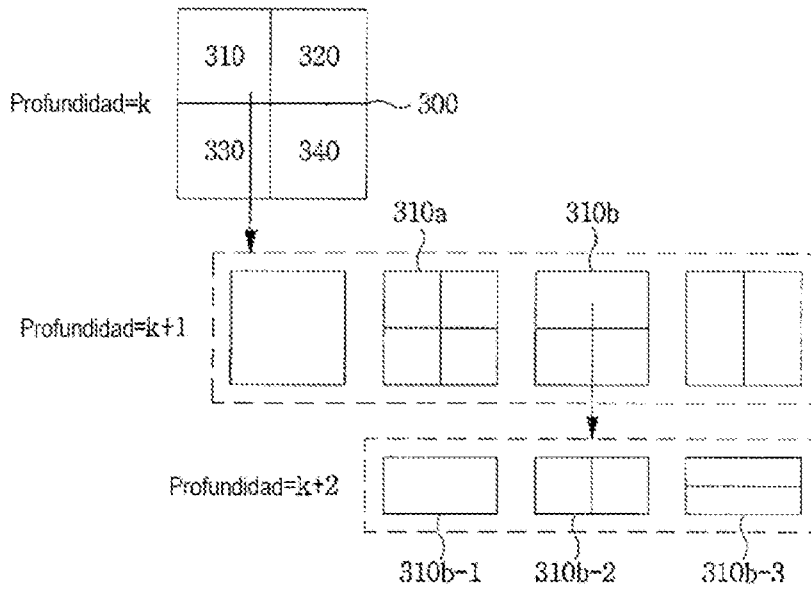
[FIG 1]



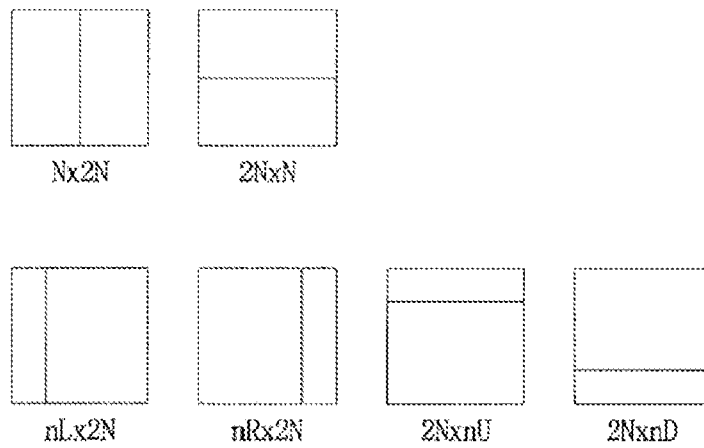
[FIG 2]



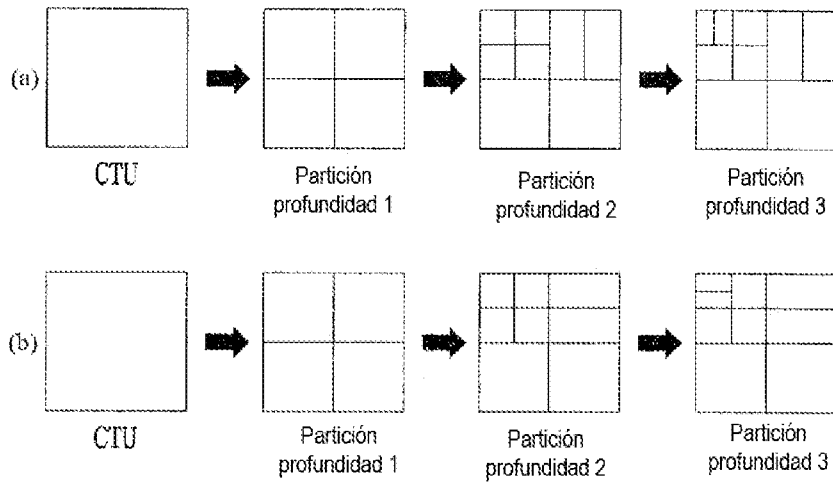
[FIG 3]



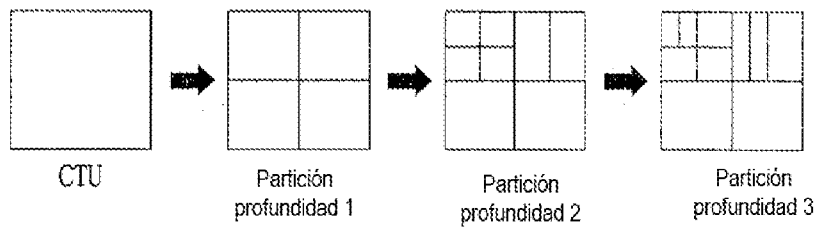
[FIG 4]



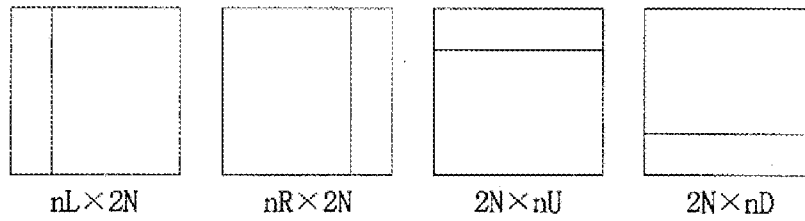
[FIG 5]



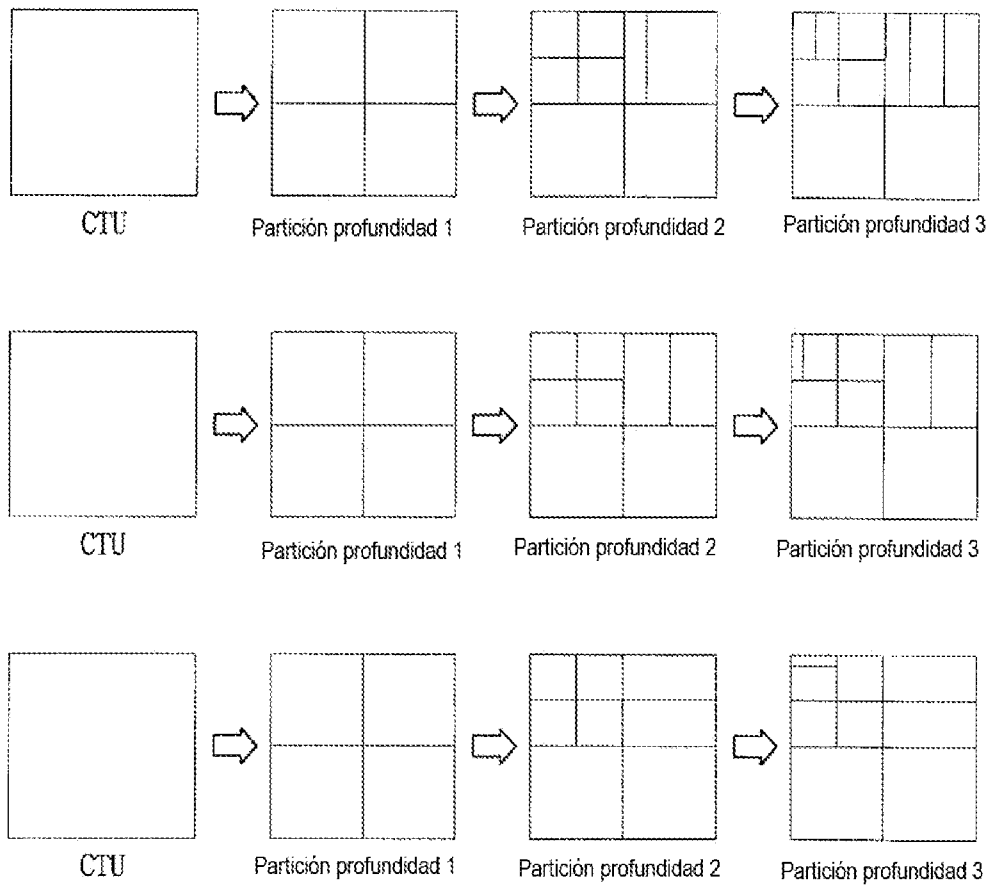
[FIG 6]



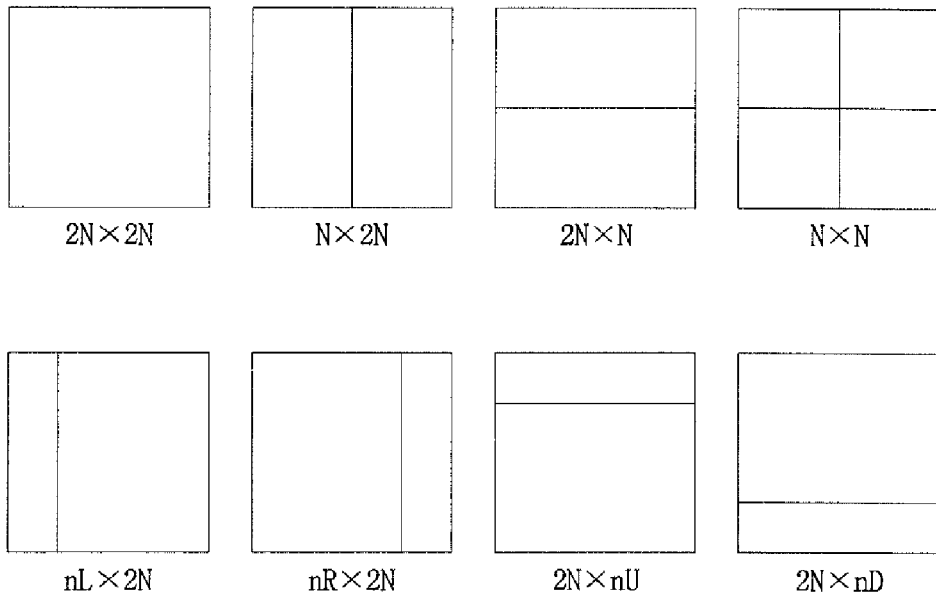
[FIG 7]



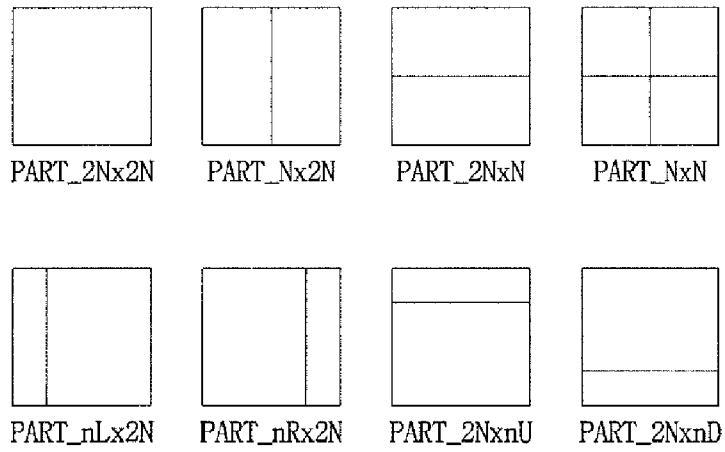
[FIG 8]



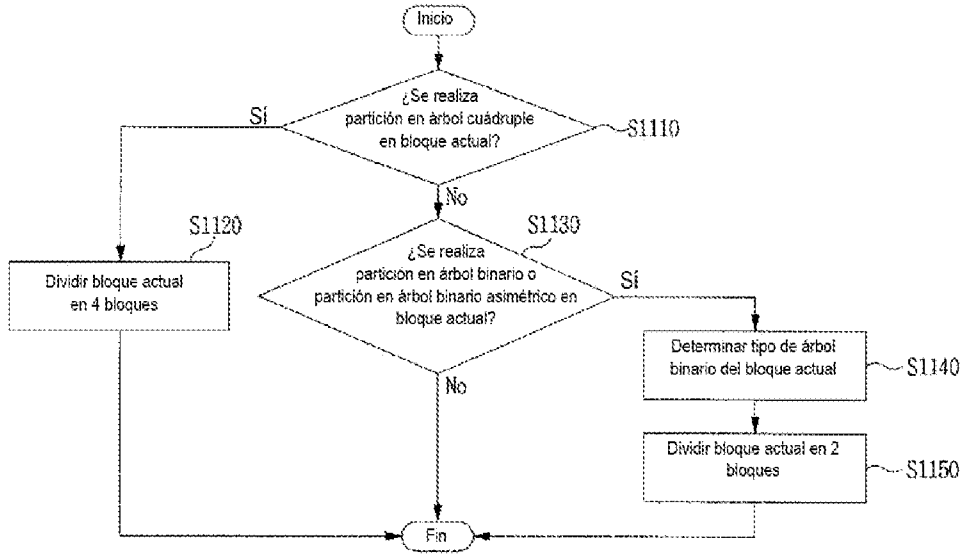
[FIG 9]



[FIG 10]



[FIG 11]



[FIG 12]

