

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 994 011**

51 Int. Cl.:

<b>H04N 19/11</b>	(2014.01)
<b>H04N 19/593</b>	(2014.01)
<b>H04N 19/176</b>	(2014.01)
<b>H04N 19/105</b>	(2014.01)
<b>H04N 19/132</b>	(2014.01)
<b>H04N 19/186</b>	(2014.01)
<b>H04N 19/463</b>	(2014.01)
<b>H04N 19/119</b>	(2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 86 Fecha de presentación y número de la solicitud internacional: **01.04.2019 PCT/KR2019/003776**
- 87 Fecha y número de publicación internacional: **10.10.2019 WO19194484**
- 96 Fecha de presentación y número de la solicitud europea: **01.04.2019 E 19780794 (4)**
- 97 Fecha y número de publicación de la concesión europea: **23.10.2024 EP 3799429**

54 Título: **Método y dispositivo de codificación/decodificación de imágenes que usa intrapredicción**

30 Prioridad:

**01.04.2018 KR 20180037811**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:  
**15.01.2025**

73 Titular/es:

**GUANGDONG OPPO MOBILE  
TELECOMMUNICATIONS CORP., LTD. (100.00%)  
No. 18, Haibin Road  
Wusha, Chang'an Dongguan Guangdong 523860,  
CN**

72 Inventor/es:

**KIM, KI BAEK**

74 Agente/Representante:

**DEL VALLE VALIENTE, Sonia**

ES 2 994 011 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**

Método y dispositivo de codificación/decodificación de imágenes que usa intrapredicción

5 **Campo técnico**

La presente invención se refiere a un método y aparato de codificación/decodificación de imágenes.

10 **Antecedentes**

Recientemente, la demanda de imágenes de alta resolución y alta calidad, tales como imágenes de alta definición (HD) e imágenes de ultra alta definición (UHD), está aumentando en diversos campos y, en consecuencia, se están analizando técnicas de compresión de imágenes de alta eficiencia.

15 Existen diversas tecnologías, tales como la tecnología de interpredicción que predice los valores de píxeles incluidos en una imagen actual a partir de una imagen anterior o posterior a una imagen actual usando tecnología de compresión de vídeo, la tecnología de intrapredicción que predice los valores de píxeles incluidos en una imagen actual usando información de píxeles en una imagen actual, una tecnología de codificación por entropía que asigna un código corto a un valor con una frecuencia de aparición alta y un código largo a un valor con una frecuencia de aparición baja. Los datos de imagen pueden comprimirse eficientemente usando tal tecnología de compresión de imágenes y transmitirse o almacenarse.

25 Alshina (Samsung) E y col.: “Description of Exploration Experiments on Coding Tools”, 116. En la reunión de MPEG, Chengdu, n.º m39573, 26 de diciembre de 2016, se describen los experimentos de exploración (EE) que se planifican realizar entre las reuniones de JVET-D y JVET-E para comprender mejor las tecnologías que se consideran para su inclusión en la próxima versión del JEM, analizar y verificar su rendimiento, complejidad e interacción con las herramientas JEM existentes. Uno de los EE se refiere al uso del método de modos directos múltiples (MDM) para la intracodificación de croma, en el que, para un bloque de croma, se selecciona un modo de una lista de modos de intrapredicción que consiste en un modo de modelo lineal de componentes cruzados, múltiples modos de intrapredicción derivados de bloques de codificación de luma cúbicos y modos de predicción de croma de bloques vecinos espaciales, añadiendo a la lista modos de cinco subbloques cúbicos. El documento US 2018/048889 A1 describe que una pluralidad de modos derivados (DM) disponibles para predecir un bloque de luma de datos de vídeo también pueden estar disponibles para predecir un bloque de croma de los datos de vídeo, correspondiendo el bloque de croma al bloque de luma, para formar una lista de candidatos de modos de predicción con respecto al bloque de croma.

35 **Resumen**

Problema técnico

40 Un objeto de la presente invención es proporcionar un método y un aparato para derivar un modo de intrapredicción.

Un objeto de la presente invención es proporcionar un método y un aparato de intrapredicción según los tipos de componentes.

45 Un objeto de la presente invención es proporcionar un método y un aparato de división de bloques para la intrapredicción.

Solución técnica

50 Según un aspecto, se proporciona un método de decodificación de una imagen como se establece en la reivindicación 1. En las reivindicaciones 2 a 8 se establecen características opcionales.

Según otro aspecto, se proporciona un método de codificación de una imagen como se expone en la reivindicación 9.

55 Según otro aspecto, se proporciona un decodificador como se expone en la reivindicación 10.

Según otro aspecto, se proporciona un codificador como se establece en la reivindicación 11.

60 Según otro aspecto, se proporciona un medio de almacenamiento informático como se establece en la reivindicación 12.

Efectos ventajosos

65 Según la presente invención, la predicción se puede realizar de manera más precisa y eficiente derivando un modo de intrapredicción basándose en una lista de MPM.

Según la presente invención, es posible mejorar la eficiencia de la interpredicción basándose en la referencia de intercomponentes.

5 La presente invención puede mejorar la eficiencia de la codificación/decodificación intrapredicción a través de la división de bloques adaptativa.

### Descripción de los dibujos

10 La Figura 1 es un diagrama de bloques que ilustra un aparato de codificación de imágenes según una realización de la presente invención.

La Figura 2 es un diagrama de bloques que ilustra un aparato de decodificación de imágenes según una realización ilustrativa de la presente invención.

15 La Figura 3 ilustra un método de intrapredicción como una realización a la que se aplica la presente invención.

La Figura 4 ilustra un método de intrapredicción en unidades de subbloques como una realización a la que se aplica la presente invención.

20 La Figura 5 ilustra un método de predicción basado en referencias de intercomponentes en una realización a la que se aplica la presente descripción.

### Mejor modo para la invención

25 El método y aparato de codificación/decodificación de imágenes de la presente invención determina un modo de intrapredicción de un bloque actual y realiza una intrapredicción en un bloque actual basándose en el modo de intrapredicción determinado.

30 En el método y aparato de codificación/decodificación de imágenes de la presente invención, el modo de intrapredicción del bloque actual puede derivarse para un bloque de luma y un bloque de croma, respectivamente.

35 En el método y aparato de codificación/decodificación de imágenes de la presente invención, un modo de intrapredicción de un bloque de luma se deriva basándose en una lista de MPM (modo más probable) y un índice de MPM, y la lista de MPM puede incluir al menos uno de un modo de intrapredicción de un bloque vecino (modo A), modo A+n, modo A-n o un modo predeterminado.

40 En el método y aparato de codificación/decodificación de imágenes de la presente invención, el bloque actual es uno cualquiera de una pluralidad de bloques de codificación inferior que pertenecen a un bloque de codificación superior, y una pluralidad de bloques de codificación inferior que pertenecen al bloque de codificación superior pueden compartir la lista de MPM.

45 Un método y aparato de codificación/decodificación de imágenes de la presente invención especifica una región de luma para la referencia de intercomponentes de un bloque de croma, realiza un submuestreo para la región de luma, deriva un parámetro para la referencia de intercomponentes del bloque de croma y predice el bloque de croma basándose en el bloque de luma submuestreado y el parámetro.

50 En el método y aparato de codificación/decodificación de imágenes de la presente invención, el bloque actual se divide en una pluralidad de subbloques, y la división puede realizarse basándose en al menos uno de un tamaño o una forma del bloque actual. La invención se expone en el conjunto de reivindicaciones adjuntas.

### Descripción detallada

55 La presente invención puede cambiarse y modificarse de manera diversa e ilustrarse con referencia a diferentes realizaciones ilustrativas, algunas de las cuales se describirán y mostrarán en los dibujos. Sin embargo, estas realizaciones no pretenden limitar la invención, sino que se interpretan en el sentido de que incluyen todas las modificaciones, equivalentes y sustituciones que pertenecen al espíritu y al alcance técnico de la invención. Los números de referencia similares en los dibujos se refieren a elementos similares en su totalidad.

60 Aunque pueden usarse los términos primero, segundo, etc., para describir diversos elementos, estos elementos no deben limitarse por estos términos. Estos términos solo se usan para distinguir un elemento de otro elemento. Por ejemplo, un primer elemento podría denominarse un segundo elemento, y un segundo elemento podría denominarse un primer elemento, de la misma manera, sin apartarse de las enseñanzas de la presente invención. El término “y/o” incluye todas y cada una de las combinaciones de una pluralidad de elementos indicados asociados.

65 Se entenderá que, cuando se menciona que un elemento está “conectado a” o “acoplado a” a otro elemento, el elemento puede estar directamente conectado o acoplado al otro elemento o puede haber elementos intermedios. Por

el contrario, cuando se menciona que un elemento está “directamente conectado a” o “directamente acoplado a” a otro elemento, no hay elementos intermedios presentes.

La terminología usada en la presente memoria tiene el propósito de describir realizaciones particulares únicamente y no pretende ser limitante de la descripción. Como se usa en la presente memoria, se pretende que las formas en singular “un”, “una”, “el” y “la” incluyan las formas en plural también, a menos que el contexto indique claramente lo contrario. Se entenderá además que los términos “incluye” y/o “tiene”, cuando se usan en esta memoria descriptiva, especifican la presencia de características, elementos integrantes, etapas, operaciones, elementos y/o componentes establecidos, pero no excluyen la presencia o adición de una o más otras características, elementos integrantes, etapas, operaciones, elementos, componentes y/o grupos de los mismos.

A continuación en la memoria, se describirán en detalle realizaciones ilustrativas de la invención con referencia a los dibujos adjuntos. Los números de referencia similares en los dibujos se refieren a elementos similares en todas partes, y las descripciones redundantes de elementos similares se omitirán en la presente memoria.

La Figura 1 es un diagrama de bloques que ilustra un aparato de codificación de imágenes según una realización de la presente invención.

Haciendo referencia a la Figura 1, el aparato de codificación de imágenes 100 incluye una unidad 110 de división de imagen, unidades 120 y 125 de predicción, una unidad 130 de transformada, una unidad 135 de cuantificación, una unidad 160 de reordenación, una unidad 165 de codificación por entropía, una unidad 140 de cuantificación inversa, una unidad 145 de transformada inversa, una unidad 150 de filtro, y una memoria 155.

Cada uno de los elementos mostrados en la Figura 1 se muestra de forma independiente para representar diferentes funciones características en el aparato de codificación, y no significa que cada elemento esté compuesto por hardware separado o un elemento de software. Es decir, los elementos están dispuestos de forma independiente por comodidad de la descripción, en donde al menos dos elementos pueden combinarse en un único elemento, o un único elemento puede dividirse en una pluralidad de elementos para realizar funciones. Cabe señalar que, las realizaciones en las que algunos elementos están integrado en un elemento combinado y/o un elemento se divide en múltiples elementos separados se incluyen en el alcance de la presente invención sin apartarse de la esencia de la presente invención.

Algunos elementos no son esenciales para las funciones sustanciales de la invención y pueden ser constituyentes opcionales para simplemente mejorar el rendimiento. La invención puede realizarse incluyendo únicamente los elementos constituyentes esenciales para la realización de la invención, excepto los elementos constituyentes usados simplemente para mejorar el rendimiento. La estructura que incluye únicamente los elementos constituyentes esenciales, excepto los constituyentes ópticos usados simplemente para mejorar el rendimiento, pertenece al alcance de la invención.

La unidad 110 de división de imagen puede dividir una imagen de entrada en al menos una unidad de procesamiento. En este caso, la unidad de procesamiento puede ser una unidad de predicción (UP), una unidad de transformada (UT) o una unidad de codificación (UC). La unidad 110 de división de imagen puede dividir una imagen en una pluralidad de combinaciones de una unidad de codificación, una unidad de predicción, y una unidad de transformación, y seleccionar una combinación de una unidad de codificación, una unidad de predicción, y una unidad de transformación basándose en un criterio predeterminado (por ejemplo, una función de coste) para codificar la imagen.

Por ejemplo, una imagen puede dividirse en una pluralidad de unidades de codificación. Para dividir una imagen en las unidades de codificación, se puede usar una estructura de árbol recursiva, tal como una estructura de árbol cuaternario. Una imagen o un bloque de codificación máximo (unidad de codificación más grande) como raíz puede dividirse en otras unidades de codificación y puede dividirse con tantos nodos secundarios como el número de unidades de codificación divididas. Una unidad de codificación que ya no se divide más según ciertas restricciones se convierte en un nodo hoja. Es decir, cuando se supone que solo es posible la división cuadrada para una unidad de codificación, una unidad de codificación puede dividirse en hasta cuatro unidades de codificación diferentes.

En las realizaciones de la invención, una unidad de codificación puede usarse para referirse no únicamente a una unidad de codificación, sino también a una unidad de decodificación.

La unidad de predicción puede ser un bloque dividido en una forma tal como al menos un cuadrado o un rectángulo del mismo tamaño dentro de una unidad de codificación, o una unidad de predicción entre las unidades de predicción divididas dentro de una unidad de codificación puede tener una forma y/o un tamaño diferentes de los de otra unidad de predicción.

Cuando una unidad de predicción que realiza una predicción intra basándose en una unidad de codificación no es una unidad de codificación mínima, la intrapredicción puede realizarse sin dividirse en una pluralidad de unidades de predicción NxN.

Las unidades 120 y 125 de predicción pueden incluir una unidad 120 de interpredicción que realiza la interpredicción y una unidad 125 de intrapredicción que realiza la intrapredicción. Las unidades 120 y 125 de predicción pueden determinar cuál de entre la interpredicción y la intrapredicción se realiza en una PU, y pueden determinar información específica (por ejemplo, un modo de intrapredicción, un vector de movimiento y una imagen de referencia) del método de predicción determinado. En este punto, una unidad de procesamiento en la que se realiza la predicción puede ser diferente de una unidad de procesamiento para la que se determinan un método de predicción e información específica en la misma. Por ejemplo, un método de predicción y un modo de predicción se pueden determinar para cada PU, mientras que puede realizarse predicción para cada TU. En la unidad 130 de transformada puede introducirse un valor residual (un bloque residual) entre un bloque previsto generado y un bloque original. Además, la información de modo de predicción, la información de vector de movimiento y similares usadas para la predicción pueden codificarse junto con el valor residual mediante la unidad 165 de codificación por entropía y transmitirse al aparato de decodificación. Cuando se usa un modo de codificación específico, el bloque original puede codificarse y transmitirse al aparato de decodificación sin generar un bloque de predicción por las unidades 120 y 125 de predicción.

La unidad 120 de interpredicción puede predecir una PU basándose en la información de al menos una imagen entre una imagen anterior de una imagen actual y una imagen posterior de una imagen actual. En algunos casos, la unidad 120 de interpredicción puede predecir una PU basándose en la información de una región parcialmente codificada en la imagen actual. La unidad 120 de interpredicción puede incluir una unidad de interpolación de imagen de referencia, una unidad de predicción de movimiento y una unidad de compensación de movimiento.

La unidad de interpolación de imágenes de referencia puede ser suministrada con información de imágenes de referencia de la memoria 155 y puede generar información de píxeles más pequeños o iguales a píxeles enteros de una imagen de referencia. En el caso de píxeles de luma, puede usarse un filtro de interpolación de 8 derivaciones basado en DCT con un coeficiente de filtro variable para generar información de píxeles más pequeños o iguales a un píxel entero en una unidad de 1/4 de píxel. En el caso de píxeles de crominancia, puede usarse un filtro de interpolación de 4 derivaciones basado en DCT con un coeficiente de filtro variable para generar información de píxeles más pequeños o iguales que un píxel entero en una unidad de 1/8 de píxel.

La unidad de predicción de movimiento puede realizar una predicción de movimiento basándose en la imagen de referencia interpolada por la unidad de interpolación de imágenes de referencia. Se pueden usar diversos métodos, tales como un algoritmo de adaptación de bloques basado en búsqueda completa (FBMA), un algoritmo de búsqueda en tres etapas (TSS) y un nuevo algoritmo de búsqueda en tres etapas (NTS) para calcular un vector de movimiento. Un vector de movimiento tiene un valor de vector de movimiento en la unidad de 1/2 o 1/4 de píxel basándose en un píxel interpolado. La unidad de predicción de movimiento puede predecir una PU actual usando diferentes métodos de predicción de movimiento. Como el modo de predicción de movimiento pueden usarse diversos métodos, tales como un modo de omisión, un modo de fusión, un modo de predicción avanzada de vectores de movimiento (AMVP), un modo de copia intrabloque, etc.

La unidad 125 de intrapredicción puede generar una PU basándose en la información sobre un píxel de referencia vecino a un bloque actual. Cuando un píxel de referencia es un píxel para el que se ha realizado la interpredicción porque un bloque vecino a la PU actual es un bloque para el que se ha realizado la interpredicción, la información sobre un píxel de referencia en el bloque para el que se ha realizado la interpredicción puede sustituirse por información sobre un píxel de referencia en un bloque para el que se ha realizado la intrapredicción. Es decir, cuando un píxel de referencia no está disponible, la información sobre el píxel de referencia no disponible puede sustituirse por información sobre al menos un píxel de referencia de los píxeles de referencia disponibles.

Un modo de predicción de intrapredicción incluye un modo de predicción direccional en el que la información de píxeles de referencia se usa según una dirección de predicción y un modo de predicción no direccional en el que la información sobre la dirección no se usa para realizar la predicción. Un modo para predecir la información de luma y un modo para predecir la información de croma pueden ser diferentes entre sí. Además, se puede usar la información del modo de intrapredicción usada para predecir la información de luma o la información de la señal de luma prevista para predecir la información de croma.

Cuando el tamaño de la unidad de predicción y el tamaño de la unidad de transformada son los mismos al realizar la intrapredicción, la intrapredicción para la unidad de predicción puede realizarse basándose en un píxel de la izquierda, un píxel de la parte superior izquierda y un píxel de la parte superior de la unidad de predicción. Sin embargo, cuando el tamaño de la unidad de predicción y el tamaño de la unidad de transformada son diferentes al realizar la intrapredicción, la intrapredicción puede realizarse usando un píxel de referencia determinado basado en la unidad de transformada. También, puede usarse la intrapredicción usando división  $N \times N$  para únicamente la unidad de codificación mínima.

En el método de intrapredicción, puede generarse un bloque previsto después de aplicar un filtro de intrasuavizado adaptativo (AIS) a píxeles de referencia según el modo de predicción. Se pueden aplicar diferentes tipos de filtros AIS a los píxeles de referencia. En el método de intrapredicción, el modo de intrapredicción de una PU actual puede predecirse a partir de un modo de intrapredicción de una PU vecina a la PU actual. Al predecir el modo de predicción de la PU actual usando la información de modo prevista desde una PU vecina, cuando la PU actual y la PU vecina

tienen el mismo modo de intrapredicción, la información que indica que la PU actual y la PU vecina tienen el mismo modo de predicción puede transmitirse usando información de bandera predeterminada. Cuando la PU actual y la PU vecina tienen diferentes modos de predicción, la información sobre el modo de predicción del bloque actual puede codificarse mediante codificación por entropía.

5 Se puede generar un bloque residual que incluye información residual. La información residual es una diferencia entre una unidad prevista generada por las unidades 120 y 125 de predicción y un bloque original de la unidad de predicción. El bloque residual generado puede introducirse a la unidad 130 de transformada.

10 La unidad 130 de transformada puede transformar el bloque residual que incluye la información residual entre la unidad prevista generada por las unidades 120 y 125 de predicción y el bloque original usando un tipo de transformada tal como DCT (transformada discreta de coseno), DST (transformada discreta de seno) o KLT. Puede determinarse si aplicar TDC, TDS o TKL para transformar el bloque residual basándose en información de modo de intrapredicción de la unidad de predicción usada para generar el bloque residual.

15 La unidad 135 de cuantificación puede cuantificar los valores transformados en un dominio de frecuencia por la unidad 130 de transformada. Un coeficiente de cuantificación puede cambiarse dependiendo de un bloque o de la importancia de una imagen. Los valores emitidos desde la unidad 135 de cuantificación pueden proporcionarse a la unidad 140 de cuantificación inversa y a la unidad 160 de reordenamiento.

20 La unidad 160 de reordenamiento puede realizar el reordenamiento de los valores de los coeficientes para el residuo cuantificado.

25 La unidad 160 de reordenamiento puede cambiar los coeficientes de un bloque bidimensional (2D) en coeficientes de un vector unidimensional (1D) a través del método de exploración de coeficientes. Por ejemplo, la unidad 160 de reordenamiento puede escanear un coeficiente de CC a un coeficiente en la región de alta frecuencia usando el método de escaneo en zig-zag y cambiarlo a una forma vectorial unidimensional. Dependiendo del tamaño de la unidad de transformada y del modo de intrapredicción, en lugar de un escaneo en zigzag, se puede usar un escaneo vertical que escanea un coeficiente de forma de bloque bidimensional en una dirección de columna y un escaneo horizontal que escanea un coeficiente de forma de bloque bidimensional en una dirección de fila. Es decir, según el tamaño de la unidad de transformada y el modo de intrapredicción, es posible determinar cuál de entre un escaneo en zigzag, un escaneo en dirección vertical y un escaneo en dirección horizontal se va a usar.

30 La unidad 165 de codificación por entropía puede realizar una codificación por entropía basándose en los valores obtenidos por la unidad 160 de reordenamiento. Se pueden usar diversos métodos de codificación, tales como la codificación de Golomb exponencial, la codificación de longitud variable adaptativa al contexto (CAVLC) o la codificación aritmética binaria adaptativa al contexto (CABAC), para la codificación por entropía.

35 La unidad 165 de codificación por entropía puede codificar una diversidad de información, tal como información de coeficiente residual e información de tipo de bloque de una unidad de codificación, información de modo de predicción, información de unidad de división, información de unidad de predicción, información de unidad de transferencia, información de vector de movimiento, información de marco de referencia, información de interpolación de bloques e información de filtrado, procedente de la unidad 160 de reordenamiento y de las unidades 120 y 125 de predicción.

40 La unidad 165 de codificación por entropía puede codificar por entropía los coeficientes de una entrada de CU desde la unidad 160 de reordenamiento.

45 La unidad 140 de cuantificación inversa y la unidad 145 de transformada inversa descuantifican los valores que se cuantifican por la unidad 135 de cuantificación y transforman inversamente los valores que se transforman por la unidad 130 de transformada. Se puede generar un bloque reconstruido añadiendo los valores residuales a la PU prevista. Los valores residuales pueden generarse mediante la unidad 140 de cuantificación inversa y la unidad 145 de transformada inversa. La PU prevista puede predecirse mediante la unidad de predicción de vector de movimiento, la unidad de compensación de movimiento y la unidad de intrapredicción de las unidades 120 y 125 de predicción.

50 La unidad 150 de filtro puede incluir al menos uno de un filtro de desbloqueo, una unidad de desplazamiento y un filtro en bucle adaptativo (ALF).

55 El filtro de desbloqueo puede eliminar la distorsión de bloque generada por los límites entre los bloques en la imagen reconstruida. Puede determinarse si aplicar el filtro de desbloqueo a un bloque actual basándose en píxeles incluidos en varias filas o columnas en el bloque. Cuando el filtro de desbloqueo se aplica a un bloque, puede aplicarse un filtro fuerte o un filtro débil dependiendo de la intensidad de filtrado de desbloqueo requerida. Cuando se realiza filtrado horizontal y filtrado vertical al aplicar el filtro de desbloqueo, puede realizarse el filtrado horizontal y el filtrado vertical en paralelo.

60 La unidad de desplazamiento puede aplicar el desplazamiento con respecto a la imagen original a la imagen filtrada de desbloqueo, en unidades de píxeles. Una región a la que se puede aplicar el desplazamiento puede determinarse después de dividir los píxeles de una imagen en un número predeterminado de regiones. El desplazamiento se puede

aplicar a la región determinada teniendo en cuenta la información de borde de cada píxel o el método de aplicación del desplazamiento a la región determinada.

5 El ALF puede realizar un filtrado basándose en un resultado de comparación de la imagen reconstruida filtrada y la imagen original. Los píxeles incluidos en una imagen se pueden dividir en grupos predeterminados, se puede determinar un filtro que se va a aplicar a cada grupo y se puede realizar un filtrado diferencial para cada grupo. Para cada unidad de codificación (UC) puede transferirse una información sobre si aplicar el ALF, y puede variar una forma y los coeficientes de filtro de un ALF que se va a aplicar a cada bloque. Además, se puede aplicar un ALF con la misma forma (forma fija) a un bloque independientemente de las características del bloque.

10 La memoria 155 puede almacenar un bloque o imagen reconstruido emitido desde la unidad de filtro 150, y el bloque o imagen reconstruido almacenado puede suministrarse a las unidades 120 y 125 de predicción cuando se realiza la interpredicción.

15 La Figura 2 es un diagrama de bloques que ilustra un aparato de decodificación de imágenes según una realización ilustrativa de la presente invención.

20 En la Figura 2, el aparato 200 de decodificación de imágenes puede incluir una unidad 210 de decodificación por entropía, una unidad 215 de reordenamiento, una unidad 220 de descuantificación, una unidad 225 de transformada inversa, unidades 230 y 235 de predicción, una unidad 240 de filtro y una memoria 245.

Cuando se introduce un flujo de bits de imagen desde el aparato de codificación de imágenes, el flujo de bits de entrada puede decodificarse en un procedimiento opuesto al del aparato de codificación de imágenes.

25 La unidad 210 de decodificación por entropía puede realizar una decodificación por entropía en un procedimiento opuesto al de realizar la codificación por entropía en la unidad de decodificación por entropía del aparato de codificación de imágenes. Por ejemplo, se pueden aplicar diversos métodos, tales como la codificación exponencial de Golomb, CAVLC o CABAC, en correspondencia al método realizado por el aparato de codificación de imágenes.

30 La unidad 210 de decodificación por entropía puede decodificar información asociada con la intrapredicción y la interpredicción realizada por el aparato de codificación.

35 La unidad 215 de reordenamiento puede realizar un reordenamiento en el flujo de bits decodificado por entropía por la unidad 210 de decodificación por entropía basándose en el método de reordenamiento del aparato de codificación. La unidad 215 de reordenamiento puede reconstruir y reordenar los coeficientes de un vector 1D en coeficientes de un bloque 2D. La unidad 215 de reordenamiento puede estar provista de información sobre el escaneo de coeficientes realizado por el aparato de codificación y puede realizar el reordenamiento usando un método de escaneo inverso de los coeficientes, basándose en el orden de escaneo realizado por el aparato de codificación.

40 La unidad 220 de descuantificación puede realizar la descuantificación basándose en un parámetro de cuantificación proporcionado por el aparato de codificación y los coeficientes reorganizados del bloque.

45 La unidad 225 de transformada inversa puede realizar una transformada inversa, es decir, una DCT inversa, una DST inversa y una KLT inversa, con respecto a la transformada realizada por la unidad de transformada, es decir, DCT, DST y KLT en el resultado de cuantificación realizado por el aparato de codificación de imágenes. La transformada inversa puede realizarse basándose en una unidad de transmisión determinada por el aparato de codificación de imágenes. La unidad 225 de transformada inversa del aparato de decodificación de imágenes puede realizar selectivamente una técnica de transformada (p. ej., DCT, DST, KLT) según una pluralidad de piezas de información tales como un método de predicción, un tamaño de un bloque actual, y una dirección de predicción.

50 Las unidades 230 y 235 de predicción pueden generar un bloque de predicción basándose en la información para generar el bloque de predicción y la información sobre un bloque o imagen previamente decodificados proporcionados. La información para generar el bloque de predicción puede proporcionarse desde la unidad 210 de decodificación por entropía. La información sobre un bloque o una imagen previamente decodificados puede proporcionarse desde la memoria 245.

60 Como se ha descrito anteriormente, si el tamaño de la unidad de predicción y el tamaño de la unidad de transformada son los mismos cuando se realiza la intrapredicción de la misma manera que la operación del aparato de codificación de imágenes, puede realizarse la intrapredicción para la unidad de predicción basándose en el píxel en la izquierda, un píxel en la parte superior izquierda y un píxel en la parte superior de la unidad de predicción. Sin embargo, cuando el tamaño de la unidad de predicción y el tamaño de la unidad de transformada son diferentes entre sí cuando se realiza la intrapredicción, puede realizarse la intrapredicción usando un píxel de referencia determinado basándose en la unidad de transformada. Además, puede usarse la intrapredicción usando división  $N \times N$  para únicamente la unidad de codificación mínima.

65

- Las unidades 230 y 235 de predicción puede incluir una unidad de determinación de unidad de predicción, una unidad de interpredicción y una unidad de intrapredicción. La unidad de determinación de unidad de predicción puede recibir una diversidad de información, tal como información de la unidad de predicción, información del modo de predicción de un método de intrapredicción e información relacionada con la predicción de movimiento de un método de interpredicción, etc., desde la unidad 210 de decodificación por entropía, y puede determinar una unidad de predicción para una unidad de codificación actual. La unidad de determinación de unidad de predicción puede determinar cuál de la interpredicción y la intrapredicción se realiza en la unidad de predicción. Una unidad 230 de interpredicción puede realizar la interpredicción en una unidad de predicción actual basándose en la información sobre al menos una imagen entre una imagen anterior y una imagen posterior de una imagen actual, incluyendo la unidad de predicción actual. En la presente memoria, una unidad 230 de interpredicción puede usar la información necesaria para la interpredicción para la unidad de predicción actual proporcionada desde el aparato de codificación de imágenes. La interpredicción puede realizarse basándose en la información de la región parcial reconstruida previamente en la imagen actual incluyendo la unidad de predicción actual.
- Para realizar la interpredicción, se puede determinar, en una unidad de una unidad de codificación, si el método de predicción de movimiento para una unidad de predicción incluida en una unidad de codificación correspondiente es un modo de omisión, un modo de fusión, un modo de AMVP o un modo de copia intrabloque.
- Una unidad 235 de intrapredicción puede generar un bloque de predicción basándose en la información de píxel en una imagen actual. Cuando una unidad de predicción es una unidad de predicción que se realiza intrapredicción, la intrapredicción puede realizarse basándose en la información de modo de intrapredicción de la unidad de predicción proporcionada por el aparato de codificación de imágenes. La unidad 235 de intrapredicción puede incluir un filtro de AIS (intrasuavizado adaptativo), una unidad de interpolación de píxel de referencia y un filtro de CC. El filtro AIS realiza filtrado de los píxeles de referencia de un bloque actual. El filtro AIS puede decidir si aplicar el filtro o no, dependiendo del modo de predicción para la unidad de predicción actual. El filtrado AIS se puede realizar en los píxeles de referencia del bloque actual usando el modo de predicción para la unidad de predicción y la información en el filtro de AIS proporcionada desde el aparato de codificación de imágenes. Cuando el modo de predicción para el bloque actual es un modo que no realiza un filtrado AIS, es posible que no se aplique el filtro AIS.
- Cuando el modo de predicción para la unidad de predicción indica un modo de predicción para realizar una intrapredicción basándose en los valores de píxel obtenidos interpolando los píxeles de referencia, la unidad de interpolación de píxeles de referencia puede generar píxeles de referencia en una unidad de un píxel fraccionario inferior a un píxel entero (es decir, un píxel completo) interpolando los píxeles de referencia. Cuando el modo de predicción de la unidad de predicción actual indica un modo de predicción de generación de un bloque de predicción sin interpolar los píxeles de referencia, los píxeles de referencia pueden no interpolarse. El filtro de CC puede generar un bloque de predicción a través del filtrado cuando el modo de predicción para bloque actual es el modo de CC.
- El bloque o imagen reconstruida puede proporcionarse a la unidad 240 de filtro. La unidad 240 de filtro incluye un filtro de desbloqueo, una unidad de desplazamiento y un ALF.
- El aparato de codificación de imágenes puede proporcionar información sobre si el filtro de desbloqueo se aplica a un correspondiente bloque o imagen, e información sobre si se aplica un filtro fuerte o un filtro débil cuando se usa el filtro de desbloqueo. El filtro de desbloqueo del aparato de codificación de imágenes puede proporcionarse con información sobre el filtro de desbloqueo desde el aparato de codificación de imágenes, y puede realizar un filtrado de desbloqueo en un bloque correspondiente.
- La unidad de desplazamiento puede aplicar un desplazamiento a la imagen reconstruida basándose en la información sobre un tipo de desplazamiento y un valor de desplazamiento aplicados a la imagen en el proceso de codificación.
- El ALF puede aplicarse a una unidad de codificación basándose en una información sobre se aplica el ALF, una información de coeficiente de ALF y similares que se proporcionan desde el aparato de codificación. La información de ALF puede incluirse y proporcionarse en un conjunto de parámetros específico.
- La memoria 245 puede almacenar el bloque o imagen reconstruido para su uso como una imagen de referencia o un bloque de referencia, y puede proporcionar la imagen reconstruida a una unidad de salida.
- Como se ha descrito anteriormente, en una realización de la presente invención, por conveniencia de descripción, se usa una unidad de codificación como unidad de codificación, pero puede ser una unidad que realiza no únicamente la codificación, sino también la decodificación.
- La Figura 3 ilustra un método de intrapredicción como una realización a la que se aplica la presente invención.
- En la Figura 3, se puede determinar un modo de intrapredicción de un bloque actual (S300).
- El bloque actual es un concepto que incluye un bloque de luma y un bloque de croma, y el modo de intrapredicción puede determinarse para cada uno de un bloque de luma y un bloque de croma. A continuación en la memoria, se

supone que el modo de intrapredicción predefinido en el aparato de decodificación está compuesto por un modo no direccional (modo plano, modo de CC) y 65 modos direccionales.

1. Derivación de modo de intrapredicción de bloque de luma

El modo de intrapredicción del bloque de luma puede derivarse basándose en una lista de MPM (modo más probable) y un índice de MPM. La lista de MPM incluye una pluralidad de MPM, y el MPM puede determinarse basándose en un modo de intrapredicción de un bloque vecino al bloque actual. El número de MPM es  $r$ , y  $r$  puede ser un número entero de 3, 4, 5, 6 o más.

Por ejemplo, la lista de MPM puede incluir al menos uno de un modo de intrapredicción de un bloque vecino (modo A), modo A-n, modo A+n o un modo predeterminado. El valor  $n$  puede ser un número entero de 1, 2, 3, 4 o mayor. El bloque vecino puede significar un bloque adyacente a la izquierda y/o en la parte superior del bloque actual. El modo predeterminado puede ser al menos uno de un modo plano, un modo de CC o un modo direccional predeterminado. El modo direccional predeterminado puede incluir al menos uno de un modo horizontal (modo V), un modo vertical (modo H), un modo V-k, un modo V+k, un modo H-k o un modo H+k.

El índice de MPM puede especificar el mismo MPM que el modo de intrapredicción del bloque de luma entre los MPM de la lista de MPM. Es decir, el MPM especificado por el índice de MPM puede establecerse como el modo de intrapredicción del bloque de luma.

Mientras tanto, la lista de MPM puede generarse basándose en un bloque de codificación superior predeterminado, y los bloques de codificación inferior que pertenecen al bloque de codificación superior pueden compartir la lista de MPM. En este punto, el bloque de codificación inferior puede corresponder al bloque actual descrito anteriormente o bloque de luma. En este caso, el índice de MPM puede señalizarse en unidades de bloques de codificación superiores. En este caso, un bloque de codificación inferior que pertenece al bloque de codificación superior puede predecirse usando el mismo modo de intrapredicción. Alternativamente, el índice de MPM puede señalizarse para cada uno de los bloques de codificación inferiores. En este caso, los bloques de codificación inferiores comparten una lista de MPM, pero pueden predecirse usando modos de intrapredicción diferentes entre sí.

El bloque de codificación superior puede definirse como un bloque de un nodo que comparte la lista de MPM, y el bloque de codificación inferior puede definirse como un bloque de un nodo hoja que ya no está dividido más en bloques de codificación más pequeños.

(Realización 1) La compartición de la lista de MPM puede realizarse únicamente cuando el bloque de codificación superior es mayor o igual a un primer tamaño umbral predeterminado. En este punto, el primer tamaño umbral puede definirse como el tamaño mínimo de un bloque en el que se permite compartir la lista de MPM. Por el contrario, la compartición de la lista de MPM puede realizarse únicamente cuando el bloque de codificación superior es menor o igual a un primer tamaño de umbral predeterminado. En este punto, el tamaño umbral puede definirse como el tamaño máximo de un bloque en el que se permite compartir la lista de MPM. En este punto, el primer tamaño umbral se puede expresar en términos de ancho (W), alto (H),  $W \times H$ ,  $W \times H$  y similares, y W y H pueden ser números enteros de 8, 16, 32 o mayores.

(Ejemplo 2) La compartición de la lista de MPM puede realizarse únicamente cuando al menos uno de una pluralidad de bloques de codificación inferiores que pertenecen al bloque de codificación superior es menor o igual que un segundo tamaño de umbral predeterminado. En este punto, el segundo tamaño umbral se puede expresar en términos de ancho (W), alto (H),  $W \times H$ ,  $W \times H$  y similares, y W y H pueden ser números enteros de 4, 8, 16, 32 o más. Sin embargo, el segundo tamaño umbral se puede establecer para que sea menor o igual que el primer tamaño umbral.

(Ejemplo 3) La compartición de la lista de MPM puede realizarse de forma adaptativa teniendo en cuenta un tipo de división del bloque de codificación superior. Por ejemplo, si el bloque de codificación superior se divide basándose en un árbol cuádruple (QT), se permite compartir la lista de MPM, de cualquier otra manera, es posible que no se permita compartir la lista de MPM. Alternativamente, cuando el bloque de codificación superior se divide basándose en un árbol binario (BT), se permite compartir la lista de MPM, y cuando el bloque de codificación superior se divide basándose en un árbol ternario (TT), es posible que no se permita compartir la lista de MPM.

La compartición de la lista de MPM puede usarse selectivamente basándose en una cualquiera de las realizaciones 1 a 3 descritas anteriormente, o la compartición de la lista de MPM puede usarse selectivamente basándose en una combinación de al menos dos de las realizaciones 1 a 3.

Además, el aparato de codificación puede señalar la información acerca de si la lista de MPM se comparte, el tamaño, la forma y la profundidad de división del nodo compartido de la lista de MPM. La compartición de la lista de MPM se puede realizar basándose en la información señalizada.

2. Derivación del modo de intrapredicción del bloque de croma

El modo de intrapredicción del bloque de croma puede derivarse como se muestra en la Tabla 1 o 2 a continuación basándose en la información (intra\_chroma\_pred\_mode) señalizada por el aparato de codificación.

[Tabla 1]

5

intra_chroma_pred_mode[ xCb ][ yCb ]	IntraPredModeY[ xCb + cbWidth / 2 ][ yCb + cbHeight / 2 ]				
	0	50	18	1	X (0 <= X <= 66)
0	66	0	0	0	0
1	50	66	50	50	50
2	18	18	66	18	18
3	1	1	1	66	1
4	0	50	18	1	X

10

15

Según la Tabla 1, el modo de intrapredicción del bloque de croma puede determinarse basándose en la información señalizada y el modo de intrapredicción del bloque de luma. En la Tabla 1, el modo 66 significa un modo diagonal en la dirección superior derecha, el modo 50 significa un modo vertical, el modo 18 significa un modo horizontal y el modo 1 significa un modo de CC. Por ejemplo, cuando el valor de la información señalizada intra\_chroma\_pred\_mode es 4, el modo de intrapredicción del bloque de croma puede establecerse para que sea el mismo que el modo de intrapredicción del bloque de luma.

20

[Tabla 2]

25

intra_chroma_pred_mode[ xCb ][ yCb ]	IntraPredModeY[ xCb + cbWidth / 2 ][ yCb + cbHeight / 2 ]				
	0	50	18	1	X (0 <= X <= 66)
0	66	0	0	0	0
1	50	66	50	50	50
2	18	18	66	18	18
3	1	1	1	66	1
4	81	81	81	81	81
5	82	82	82	82	82
6	83	83	83	83	83
7	0	50	18	1	X

30

35

40

La Tabla 2 se puede aplicar cuando se permite la predicción basada en referencia de intercomponentes para un bloque de croma. La Tabla 2 soporta el modo 81, el modo 82 y el modo 83 como modos de intrapredicción de un bloque de croma, que corresponden a modos de predicción basados en referencias de intercomponentes.

45

En la Figura 3, basándose en el modo de intrapredicción derivado en S300, la intrapredicción puede realizarse en un bloque actual (S310).

50

La intrapredicción puede realizarse en unidades de subbloques del bloque actual. Para este fin, el bloque actual puede dividirse en una pluralidad de subbloques. El método de división se describirá con más detalle con referencia a la Figura 4.

55

Mientras tanto, cuando el modo de intrapredicción derivado corresponde a un modo de predicción basado en referencia de intercomponentes, se puede predecir un bloque de croma a partir de un bloque de luma construido previamente, que se describirá en detalle con referencia a la Figura 5.

60

La Figura 4 ilustra un método de intrapredicción en unidades de subbloques como una realización a la que se aplica la presente invención.

Como se ha descrito anteriormente, el bloque actual puede dividirse en una pluralidad de subbloques. En este caso, el bloque actual puede corresponder a un nodo hoja. Un nodo hoja puede significar un bloque de codificación que ya no está dividido más en bloques de codificación más pequeños.

65

La división se puede realizar basándose en el tamaño del bloque actual (Realización 1).

En la Figura 4, cuando el tamaño del bloque actual 400 es menor que un tamaño umbral predeterminado, el bloque actual puede dividirse en dos vertical u horizontalmente. Por el contrario, cuando el tamaño del bloque actual 410 es mayor o igual al tamaño umbral, el bloque actual puede dividirse en cuatro vertical u horizontalmente.

5 El tamaño umbral puede ser señalizarse por el aparato de codificación o puede ser un valor fijo predefinido en el aparato de decodificación. Por ejemplo, el tamaño umbral se expresa como  $N \times M$ , y  $N$  y  $M$  pueden ser 4, 8, 16 o más. Los valores  $N$  y  $M$  pueden ser iguales o pueden configurarse de manera diferente entre sí.

10 Alternativamente, si el tamaño del bloque actual es menor que el tamaño umbral predeterminado, el bloque actual no puede dividirse (sin división); de lo contrario, el bloque actual puede dividirse en dos o cuatro.

La división se puede realizar basándose en la forma del bloque actual (Realización 2).

15 Si la forma del bloque actual es un cuadrado, el bloque actual puede dividirse en cuatro, de cualquier otra manera, el bloque actual puede dividirse en dos. Por el contrario, cuando la forma del bloque actual es un cuadrado, el bloque actual puede dividirse en dos y, de cualquier otra manera, el bloque actual puede dividirse en cuatro.

20 Alternativamente, si la forma del bloque actual es un cuadrado, el bloque actual se divide en dos o cuatro, de cualquier otra manera, puede que el bloque actual no se divida. Por el contrario, si la forma del bloque actual es un cuadrado, el bloque actual no puede dividirse, de cualquier otra manera, el bloque actual puede dividirse en dos o cuatro.

Puede dividirse aplicando selectivamente cualquiera de las Realizaciones 1 o 2 descritas anteriormente, o puede dividirse basándose en una combinación de las realizaciones 1 y 2.

25 La división de 2 es dividir en dos en dirección vertical u horizontal, y la división en 4 puede incluir dividir en cuatro en dirección vertical u horizontal, o dividir en cuatro en direcciones vertical y horizontal.

30 En la realización anterior, se describe la división en 2 o la división en 4, pero la presente invención no está limitada a las mismas, y el bloque actual puede dividirse en tres en una dirección vertical u horizontal. En este caso, la relación entre el ancho o el alto puede ser (1:1:2), (1:2:1) o (2:1:1).

35 La información sobre si debe dividirse en unidades de subbloque, si debe dividirse en cuatro, una dirección de división y el número de divisiones puede señalizarse por el aparato de codificación o determinarse de forma variable por el aparato de decodificación basándose en un parámetro de codificación predeterminado. En este punto, el parámetro de codificación es representativo de un tamaño/forma de bloque, un tipo de división (división en 4, división en 2, división en 3), un modo de intrapredicción, un rango/posición de píxeles vecinos para la intrapredicción y unos tipos de componentes (p. ej., luma y croma), un tamaño máximo/mínimo de un bloque de transformada, un tipo de transformada (p. ej., salto de transformada, DCT2, DST7, DCT8), y similares.

40 La Figura 5 ilustra un método de predicción basado en referencias de intercomponentes en una realización a la que se aplica la presente descripción.

45 El bloque actual puede dividirse en un bloque de luma y un bloque de croma según el tipo de componente. El bloque de croma se puede predecir usando el píxel del bloque de luma reconstruido. Esto se denomina referencia de intercomponentes. En esta realización, se supone que el bloque de croma tiene un tamaño de  $(nTbW \times nTbH)$ , y que el bloque de luma correspondiente al bloque de croma tiene un tamaño de  $(2 \times nTbW \times 2 \times nTbH)$ .

50 En la Figura 5, se puede especificar una región de luma para la referencia intercomponentes del bloque de croma (S500).

55 La región de luma puede incluir al menos una de un bloque de luma o una región vecina adyacente al bloque de luma. En este punto, el bloque de luma se puede definir como una región que incluye píxeles  $pY[x][y]$  ( $x=0..nTbW \times 2 - 1$ ,  $y=0..nTbH \times 2 - 1$ ). Los píxeles pueden significar valores reconstruidos antes de que se aplique el filtro en bucle.

60 La región vecina puede incluir al menos una de una región vecina izquierda, una región vecina superior o una región vecina superior izquierda. La región vecina izquierda se puede establecer como una región que incluye los píxeles  $pY[x][y]$  ( $x=-1..-3$ ,  $y=0..2 \times \text{numSampL} - 1$ ). El ajuste únicamente se puede realizar cuando el valor de  $\text{numSampL}$  es mayor que 0. La región vecina superior se puede establecer como una región que incluye los píxeles  $pY[x][y]$  ( $x=0..2 \times \text{numSampT} - 1$ ,  $y=-1..-3$ ). El ajuste únicamente se puede realizar cuando el valor de  $\text{numSampT}$  es mayor que 0. La región vecina superior izquierda se puede establecer como una región que incluye los píxeles  $pY[x][y]$  ( $x=-1$ ,  $y=-1..-2$ ). El ajuste únicamente se puede realizar cuando la región superior izquierda del bloque de luma está disponible.

65 Los  $\text{numSampL}$  and  $\text{numSampT}$  descritos anteriormente pueden determinarse basándose en el modo de intrapredicción del bloque actual.

Por ejemplo, cuando el modo de intrapredicción del bloque actual es INTRA\_LT\_CCLM, puede derivarse basándose en la Ecuación 1. En este caso, INTRA\_LT\_CCLM puede significar un modo en el que la referencia intercomponentes se realiza basándose en las regiones vecinas izquierda y superior al bloque actual.

5 [Ecuación 1]

$$\text{numSampT} = \zeta_{\text{availT}}? \text{nTbW}: 0$$

$$\text{numSampL} = \zeta_{\text{availL}}? \text{nTbH}: 0$$

10 Según la ecuación 1, numSampT se deriva como nTbW cuando la región vecina superior al bloque actual está disponible. De cualquier otra manera, numSampT puede derivarse como 0. Similarmente, numSampL se deriva como nTbH cuando la región vecina izquierda del bloque actual está disponible. De cualquier otra manera, numSampT puede derivarse como 0.

15 Por el contrario, cuando el modo de intrapredicción del bloque actual no es INTRA\_LT\_CCLM, puede derivarse basándose en la ecuación 2 a continuación.

20 [Ecuación 2]

$$\text{numSampT} = \zeta_{(\text{availT} \ \&\& \ \text{predModelIntra} == \text{INTRA\_T\_CCLM})} (\text{nTbW} + \text{numTopRight}): 0$$

$$\text{numSampL} = \zeta_{(\text{availL} \ \&\& \ \text{predModelIntra} == \text{INTRA\_L\_CCLM})} (\text{nTbH} + \text{numLeftBelow}): 0$$

25 En la ecuación 2, INTRA\_T\_CCLM puede referirse a un modo en el que la referencia de intercomponentes se realiza basándose en una región vecina superior al bloque actual. INTRA\_L\_CCLM puede referirse a un modo en el que la referencia de intercomponentes se realiza basándose en una región vecina izquierda al bloque actual. numTopRight puede significar el número de todos o algunos píxeles que pertenecen a una región vecina superior derecha del bloque de croma. Algunos píxeles pueden hacer referencia a los píxeles disponibles entre los píxeles que pertenecen a la fila de píxeles más baja de la región correspondiente. En una determinación de disponibilidad, si los píxeles están disponibles puede determinarse secuencialmente en una dirección de izquierda a derecha. Este proceso se puede realizar hasta que se encuentre un píxel no disponible. numLeftBelow puede significar el número de todos o algunos de los píxeles que pertenecen a una región vecina inferior izquierda del bloque de croma. Algunos píxeles pueden hacer referencia a los píxeles disponibles entre los píxeles que pertenecen a la línea de píxeles más a la derecha de la región correspondiente. En la determinación de disponibilidad, se puede determinar secuencialmente si los píxeles están disponibles en una dirección de arriba a abajo. Este proceso puede realizarse hasta que se encuentre un píxel no disponible.

Haciendo referencia a la Figura 5, el submuestreo se puede realizar en la región de luma especificada en S500 (S510).

40 El submuestreo puede incluir al menos uno de 1. El submuestreo del bloque de luma, 2. El submuestreo de la región vecina izquierda al bloque de luma, o 3. El submuestreo de la región vecina superior al bloque de luma. Esto se describirá en detalle a continuación.

45 1. El submuestreo de bloque de luma

(Realización 1)

50 El píxel pDsY[x][y] (x=0..nTbW-1, y=0..nTbH-1) del bloque de luma submuestreado puede derivarse basándose en un píxel pY[2\*x][2\*y] correspondiente del bloque de luma y el píxel vecino. El píxel vecino puede significar al menos uno de un píxel vecino izquierdo, un píxel vecino derecho, un píxel vecino superior o un píxel vecino inferior al píxel correspondiente. Por ejemplo, el píxel pDsY[x][y] puede derivarse basándose en la ecuación 3 a continuación.

$$\text{pDsY}[x][y] = (\text{pY}[2 \times x][2 \times y - 1] + \text{pY}[2 \times x - 1][2 \times y] + 4 \times \text{pY}[2 \times x][2 \times y] + \text{pY}[2 \times x + 1][2 \times y] + \text{pY}[2 \times x][2 \times y + 1] + 4) \gg 3 \quad \text{[Ecuación 3]}$$

55 Sin embargo, puede haber un caso donde la región vecina izquierda/superior al bloque actual no esté disponible. Cuando la región vecina izquierda al bloque actual no está disponible, el píxel pDsY[0][y] (y=1..nTbH-1) del bloque de luma submuestreado puede derivarse basándose en el píxel correspondiente pY[0][2\*y] del bloque de luma y el píxel vecino del píxel correspondiente. El píxel vecino puede significar al menos uno del píxel vecino superior o el píxel vecino inferior al píxel correspondiente. Por ejemplo, el píxel pDsY[0][y] (y=1..nTbH-1) puede derivarse basándose en la ecuación 4 a continuación.

$$\text{pDsY}[0][y] = (\text{pY}[0][2 \times y - 1] + 2 \times \text{pY}[0][2 \times y] + \text{pY}[0][2 \times y + 1] + 2) \gg 2 \quad \text{[Ecuación 4]}$$

65 Cuando la región vecina superior al bloque actual no está disponible, el píxel pDsY[x][0] (x=1..nTbW-1) del bloque de luma submuestreado puede derivarse basándose en el píxel correspondiente pY[2\*x][0] del bloque de luma y el píxel

vecino del píxel correspondiente. El píxel vecino puede significar al menos uno del píxel vecino izquierdo o el píxel vecino derecho al píxel correspondiente. Por ejemplo, el píxel  $pDsY[x][0]$  ( $x=1..nTbW-1$ ) puede derivarse basándose en la ecuación 5 a continuación.

$$pDsY[x][0] = (pY[2 \times x-1][0] + 2 \times pY[2 \times x][0] + pY[2 \times x+1][0] + 2) \gg 2 \quad \text{[Ecuación 5]}$$

El píxel  $pDsY[0][0]$  del bloque de luma submuestreado puede derivarse basándose en el píxel  $pY[0][0]$  correspondiente del bloque de luma y/o un píxel vecino del píxel correspondiente. La posición del píxel vecino puede variar dependiendo de si están disponibles las regiones vecinas izquierda/superior del bloque actual.

Por ejemplo, cuando la región vecina izquierda está disponible, pero la región vecina superior no está disponible,  $pDsY[0][0]$  puede derivarse basándose en la ecuación 6 a continuación.

[Ecuación 6]

$$pDsY[0][0] = (pY[-1][0] + 2 \times pY[0][0] + pY[1][0] + 2) \gg 2$$

Por el contrario, cuando la región vecina izquierda no está disponible, pero la región vecina superior está disponible,  $pDsY[0][0]$  puede derivarse basándose en la ecuación 7 a continuación.

[Ecuación 7]

$$pDsY[0][0] = (pY[0][-1] + 2 \times pY[0][0] + pY[0][1] + 2) \gg 2$$

En otro ejemplo, cuando ambas de las regiones vecinas izquierda y superior no están disponibles,  $pDsY[0][0]$  puede establecerse como el píxel  $pY[0][0]$  correspondiente del bloque de luma.

(Realización 2)

El píxel  $pDsY[x][y]$  ( $x=0..nTbW-1$ ,  $y=0..nTbH-1$ ) del bloque de luma submuestreado puede derivarse basándose en el píxel  $pY[2 \times x][2 \times y]$  correspondiente del bloque de luma y el píxel vecino del píxel correspondiente. El píxel vecino puede significar al menos uno de entre un píxel vecino inferior, un píxel vecino izquierdo, un píxel vecino derecho, un píxel vecino inferior izquierdo o un píxel vecino inferior derecho al píxel correspondiente. Por ejemplo, el píxel  $pDsY[x][y]$  puede derivarse basándose en la ecuación 8 a continuación.

$$pDsY[x][y] = (pY[2 \times x-1][2 \times y] + pY[2 \times x-1][2 \times y+1] + 2 \times pY[2 \times x][2 \times y] + 2 \times pY[2 \times x][2 \times y+1] + pY[2 \times x+1][2 \times y] + pY[2 \times x+1][2 \times y+1] + 4) \gg 3 \quad \text{[Ecuación 8]}$$

Sin embargo, cuando la región vecina izquierda al bloque actual no está disponible, el píxel  $pDsY[0][y]$  ( $y=0..nTbH-1$ ) del bloque de luma submuestreado puede derivarse basándose en el píxel correspondiente  $pY[0][2 \times y]$  del bloque de luma y un píxel vecino inferior al mismo. Por ejemplo, el píxel  $pDsY[0][y]$  ( $y=0..nTbH-1$ ) puede derivarse basándose en la ecuación 9 a continuación.

[Ecuación 9]

$$pDsY[0][y] = (pY[0][2 \times y] + pY[0][2 \times y + 1] + 1) \gg 1$$

El submuestreo del bloque de luma se puede realizar basándose en una de las realizaciones 1 y 2, como se ha descrito anteriormente. En este punto, una de las realizaciones 1 y 2 puede seleccionarse basándose en una bandera predeterminada. La bandera puede indicar si el píxel de luma submuestreado tiene la misma posición que la del píxel de luma original. Por ejemplo, cuando la bandera es un primer valor, el píxel de luma submuestreado tiene la misma posición que la del píxel de luma original. Por el contrario, cuando la bandera es un segundo valor, el píxel de luma submuestreado tiene la misma posición que la del píxel de luma original en la dirección horizontal, pero tiene una posición desplazada medio píxel en la dirección vertical.

## 2. Submuestreo de la región vecina izquierda al bloque de luma

(Realización 1)

El píxel  $pLeftDsY[y]$  ( $y=0..numSampL-1$ ) de la región vecina izquierda submuestreada puede derivarse basándose en el píxel  $pY[-2][2 \times y]$  correspondiente de la región vecina izquierda y un píxel vecino del píxel correspondiente. El píxel vecino puede significar al menos uno de un píxel vecino izquierdo, un píxel vecino derecho, un píxel vecino superior o un píxel vecino inferior al píxel correspondiente. Por ejemplo, el píxel  $pLeftDsY[y]$  se puede derivar basándose en la ecuación 10 a continuación.

$$pLeftDsY[y] = (pY[-2][2 \times y - 1] + pY[-3][2 \times y] + 4 \times pY[-2][2 \times y] + pY[-1][2 \times y] + pY[-2][2 \times y + 1] + 4) \gg 3 \quad \text{[Ecuación 10]}$$

5 Sin embargo, cuando la región vecina superior izquierda del bloque actual no está disponible, el píxel  $pLeftDsY[0]$  de la región vecina izquierda submuestreada puede derivarse basándose en el píxel correspondiente  $pY[-2][0]$  de la región vecina izquierda y un píxel vecino del píxel correspondiente. El píxel vecino puede significar al menos uno del píxel vecino izquierdo o un píxel vecino derecho al píxel correspondiente. Por ejemplo, el píxel  $pLeftDsY[0]$  puede derivarse basándose en la ecuación 11 a continuación.

10 [Ecuación 11]

$$pLeftDsY[0] = (pY[-3][0] + 2 \times pY[-2][0] + pY[-1][0] + 2) \gg 2$$

15 (Realización 2)

El píxel  $pLeftDsY[y]$  ( $y=0..numSampL-1$ ) de la región vecina izquierda submuestreada puede derivarse basándose en el píxel  $pY[-2][2 \times y]$  correspondiente de la región vecina izquierda y un píxel vecino alrededor del píxel correspondiente. El píxel vecino puede significar al menos uno de entre un píxel vecino inferior, un píxel vecino izquierdo, un píxel vecino derecho, un píxel vecino inferior izquierdo o un píxel vecino inferior derecho al píxel correspondiente. Por ejemplo, el píxel  $pLeftDsY[y]$  puede derivarse basándose en la siguiente ecuación 12.

$$pLeftDsY[y] = (pY[-1][2 \times y] + pY[-1][2 \times y + 1] + 2 \times pY[-2][2 \times y] + 2 \times pY[-2][2 \times y + 1] + pY[-3][2 \times y] + pY[-3][2 \times y + 1] + 4) \gg 3 \quad \text{[Ecuación 12]}$$

30 Similarmente, el submuestreo de la región vecina izquierda puede realizarse basándose en una de las realizaciones 1 y 2, como se ha descrito anteriormente. En este punto, una de las realizaciones 1 y 2 puede seleccionarse basándose en una bandera predeterminada. La bandera indica si el píxel de luma submuestreado tiene la misma posición que la del píxel de luma original. Esto es lo mismo que se ha descrito anteriormente.

35 El submuestreo de la región vecina izquierda únicamente se puede realizar cuando el valor de  $numSampL$  es mayor que 0. Cuando el valor de  $numSampL$  es mayor que 0, puede significar que la región vecina izquierda del bloque actual está disponible, y el modo de intrapredicción del bloque actual es  $INTRA\_LT\_CCLM$  o  $INTRA\_L\_CCLM$ .

### 3. Submuestreo de la región vecina superior al bloque de luma

40 (Realización 1)

El píxel  $pTopDsY[x]$  ( $x=0..numSampT-1$ ) de la región vecina superior submuestreada puede derivarse teniendo en cuenta si la región vecina superior pertenece a una CTU diferente de una CTU a la que pertenece el bloque de luma.

45 Cuando la región vecina superior pertenece a la misma CTU que el bloque de luma, el píxel  $pTopDsY[x]$  de la región vecina superior submuestreada puede derivarse basándose en el píxel  $pY[2 \times x][-2]$  correspondiente de la región vecina superior y un píxel vecino del píxel correspondiente. El píxel vecino puede significar al menos uno de un píxel vecino izquierdo, un píxel vecino derecho, un píxel vecino superior o un píxel vecino inferior al píxel correspondiente. Por ejemplo, el píxel  $pTopDsY[x]$  puede derivarse basándose en la ecuación 13 a continuación.

$$pTopDsY[x] = (pY[2 \times x][-3] + pY[2 \times x - 1][-2] + 4 \times pY[2 \times x][-2] + pY[2 \times x + 1][-2] + pY[2 \times x][-1] + 4) \gg 3 \quad \text{[Ecuación 13]}$$

55 Por el contrario, cuando la región vecina superior pertenece a una CTU diferente del bloque de luma, el píxel  $pTopDsY[x]$  de la región vecina superior submuestreada puede derivarse basándose en el píxel  $pY[2 \times x][-1]$  correspondiente de la región vecina superior y un píxel vecino del píxel correspondiente. El píxel vecino puede significar al menos uno del píxel vecino izquierdo o el píxel vecino derecho al píxel correspondiente. Por ejemplo, el píxel  $pTopDsY[x]$  puede derivarse basándose en la ecuación 14 a continuación.

$$pTopDsY[x] = (pY[2 \times x - 1][-1] + 2 \times pY[2 \times x][-1] + pY[2 \times x + 1][-1] + 2) \gg 2 \quad \text{[Ecuación 14]}$$

65

Alternativamente, cuando la región vecina superior izquierda del bloque actual no está disponible, el píxel vecino puede significar al menos uno del píxel vecino superior o el píxel vecino inferior del píxel correspondiente. Por ejemplo, el píxel  $pTopDsY[0]$  puede derivarse basándose en la ecuación 15 a continuación.

5 [Ecuación 15]

$$pTopDsY[0] = (pY[0][ -3] + 2 \times pY[0][ -2] + pY[0][ -1] + 2) \gg 2$$

10 Alternativamente, cuando la región vecina superior izquierda del bloque actual no está disponible y la región vecina superior pertenece a una CTU diferente del bloque de luma, el píxel  $pTopDsY[0]$  se puede establecer como el píxel  $pY[0][ -1]$  de la región vecina superior.

(Realización 2)

15 El píxel  $pTopDsY[x]$  ( $x=0..numSampT-1$ ) de la región vecina superior submuestreada puede derivarse teniendo en cuenta si la región vecina superior pertenece a una CTU diferente del bloque de luma.

20 Cuando la región vecina superior pertenece a la misma CTU que el bloque de luma, el píxel  $pTopDsY[x]$  de la región vecina superior submuestreada puede derivarse basándose en el píxel  $pY[2 \times x][ -2]$  correspondiente de la región vecina superior y un píxel vecino del píxel correspondiente. El píxel vecino puede significar al menos uno de entre un píxel vecino inferior, un píxel vecino izquierdo, un píxel vecino derecho, un píxel vecino inferior izquierdo o un píxel vecino inferior derecho al píxel correspondiente. Por ejemplo, el píxel  $pTopDsY[x]$  puede derivarse basándose en la ecuación 16 a continuación.

25 
$$pTopDsY[x] = (pY[2 \times x - 1][ -2] + pY[2 \times x - 1][ -1] + 2 \times pY[2 \times x][ -2] + 2 \times pY[2 \times x][ -1] + pY[2 \times x + 1][ -2] + pY[2 \times x + 1][ -1] + 4) \gg 3$$
 [Ecuación 16]

30 Por el contrario, cuando la región vecina superior pertenece a una CTU diferente del bloque de luma, el píxel  $pTopDsY[x]$  de la región vecina superior submuestreada puede derivarse basándose en el píxel  $pY[2 \times x][ -1]$  correspondiente de la región vecina superior y un píxel vecino del píxel correspondiente. El píxel vecino puede significar al menos uno del píxel vecino izquierdo o el píxel vecino derecho al píxel correspondiente. Por ejemplo, el píxel  $pTopDsY[x]$  puede derivarse basándose en la ecuación 17 a continuación.

35 
$$pTopDsY[x] = (pY[2 \times x - 1][ -1] + 2 \times pY[2 \times x][ -1] + pY[2 \times x + 1][ -1] + 2) \gg 2$$
 [Ecuación 17]

40 Alternativamente, cuando la región vecina superior izquierda del bloque actual no está disponible, el píxel vecino puede significar al menos uno del píxel vecino superior o el píxel vecino inferior del píxel correspondiente. Por ejemplo, el píxel  $pTopDsY[0]$  puede derivarse basándose en la ecuación 18 a continuación.

45 [Ecuación 18]

$$pTopDsY[0] = (pY[0][ -2] + pY[0][ -1] + 1) \gg 1$$

50 Alternativamente, cuando la región vecina superior izquierda del bloque actual no está disponible y la región vecina superior pertenece a una CTU diferente del bloque de luma, el píxel  $pTopDsY[0]$  se puede establecer como el píxel  $pY[0][ -1]$  de la región vecina superior.

55 De manera similar, el submuestreo de la región vecina superior puede realizarse basándose en una de las realizaciones 1 y 2, como se ha descrito anteriormente. En este punto, una de las realizaciones 1 y 2 puede seleccionarse basándose en una bandera predeterminada. La bandera indica si el píxel de luma submuestreado tiene la misma posición que la del píxel de luma original. Esto es lo mismo que se ha descrito anteriormente.

60 En un ejemplo, el submuestreo de la región vecina superior se puede realizar únicamente cuando el valor de  $numSampT$  es mayor que 0. Cuando el valor de  $numSampT$  es mayor que 0, puede significar que la región vecina superior del bloque actual está disponible, y el modo de intrapredicción del bloque actual es `INTRA_LT_CCLM` o `INTRA_T_CCLM`.

Se puede derivar un parámetro para la referencia de intercomponentes del bloque de croma (S520).

65 El parámetro puede determinarse teniendo en cuenta el modo de intrapredicción del bloque actual. El parámetro puede derivarse usando al menos uno de un píxel de la región de luma o un píxel de las regiones vecinas superior/izquierda de un bloque de croma. En este punto, la región de luma puede incluir al menos una del bloque de luma o las regiones

vecinas superior/izquierda del bloque de luma. La región de luma puede significar una región a la que se aplica el submuestreo mencionado anteriormente.

El bloque de croma se puede predecir basándose en el bloque de luma submuestreado y en el parámetro (S530).

El número de parámetros aplicados al bloque de luma puede ser k. En este caso, k puede ser un número entero de 1, 2, 3 o más. Al menos uno de los parámetros puede ser un factor de escalado aplicado a un valor de píxel del bloque de luma. Los parámetros pueden incluir además un parámetro de desplazamiento aplicado al valor de píxel o al valor de píxel escalado del bloque de luma.

Los métodos según la presente invención pueden implementarse en forma de instrucciones de programa que pueden ser ejecutadas a través de diversos medios informáticos y grabarse en un medio legible por ordenador. El medio legible por ordenador puede incluir instrucciones de programa, ficheros de datos, estructuras de datos y similares en solitario o en combinación. Las instrucciones de programa grabadas en el medio legible por ordenador pueden diseñarse y configurarse especialmente para la presente invención, o pueden ser conocidas y utilizables por los expertos en el campo del software informático.

Los ejemplos de medios legibles por ordenador pueden incluir dispositivos de hardware especialmente configurados para almacenar y ejecutar instrucciones de programa, tales como ROM, RAM, memoria flash y similares. Los ejemplos de instrucciones de programa pueden incluir códigos de lenguaje de alto nivel que pueden ser ejecutados por un ordenador usando un intérprete o similar, así como códigos de lenguaje de máquina tales como los producidos por un compilador. El dispositivo de hardware descrito anteriormente puede configurarse para funcionar como al menos un módulo de software para realizar la operación de la presente invención, y viceversa.

Además, el método o aparato descrito anteriormente puede implementarse combinando toda o parte de su configuración o función, o puede implementarse por separado.

La invención según se reivindica se establece en las siguientes reivindicaciones.

Disponibilidad industrial

La presente invención puede usarse para codificar/decodificar una imagen.

REIVINDICACIONES

1. Un método de decodificación de una imagen, que comprende:

5 determinar (S300) un modo de intrapredicción de un bloque actual (400, 410); y  
 realizar (S310) intrapredicción en el bloque actual (400, 410) basándose en el modo de intrapredicción,  
 en donde el bloque actual (400, 410) comprende un bloque de luma y un bloque de croma,  
 en donde el modo de intrapredicción comprende un modo de intrapredicción de luma para la intrapredicción del bloque de luma y un modo de intrapredicción de croma para la intrapredicción del  
 10 bloque de croma,  
 en donde el modo de intrapredicción de luma se deriva basándose en una lista de MPM y un índice de MPM (modo más probable), y  
 en donde la lista de MPM comprende al menos uno del modo A, el modo A+n o el modo A-n, en donde el modo A representa un modo de intrapredicción de un bloque vecino,  
 15 en donde realizar la intrapredicción comprende:

especificar (S500) una región de luma para la referencia de intercomponentes del bloque de croma;  
 20 realizar (S510) un submuestreo para la región de luma;  
 derivar (S520) un parámetro para la referencia de intercomponentes del bloque de croma;  
 y  
 predecir (S530) el bloque de croma basándose en la región de luma submuestreada y el parámetro,

25 en donde la región de luma comprende un bloque de luma y una región vecina adyacente al bloque de luma, y  
 realizar submuestreo para la región de luma comprende:

30 realizar submuestreo para el bloque de luma basándose en una de:

$$pDsY[x][y] = (pY[2 * x][2 * y-1] + pY[2 * x-1][2 * y] + 4 * pY[2 * x][2 * y] + pY[2 * x+1][2 * y] + pY[2 * x][2 * y+1] + 4) >> 3, \text{ o}$$

$$35 \quad pDsY[x][y] = (pY[2 * x-1][2 * y] + pY[2 * x-1][2 * y+1] + 2 * pY[2 * x][2 * y] + 2 * pY[2 * x][2 * y+1] + pY[2 * x+1][2 * y] + pY[2 * x+1][2 * y+1] + 4) >> 3$$

en donde pY[a][b] representa un valor de píxel de un píxel que tiene las coordenadas (a, b).

40 2. El método de la reivindicación 1, en donde el bloque actual (400, 410) está dividido en una pluralidad de subbloques, y

45 en donde la división se realiza basándose en un tamaño del bloque actual (400, 410), y cuando el tamaño del bloque actual (400) es menor que un tamaño umbral predeterminado, el bloque actual (400) se divide en dos vertical u horizontalmente, y cuando el tamaño del bloque actual (410) es mayor o igual al tamaño umbral predeterminado, el bloque actual (410) se divide en cuatro, ya sea vertical u horizontalmente.

50 3. El método de la reivindicación 1, en donde realizar submuestreo para la región de luma comprende: realizar submuestreo para la región vecina adyacente al bloque de luma basándose en:

$$pLeftDsY[y] = (pY[-2][2 * y-1] + pY[-3][2 * y] + 4 * pY[-2][2 * y] + pY[-1][2 * y] + pY[-2][2 * y+1] + 4) >> 3.$$

55 4. El método de la reivindicación 1, en donde realizar submuestreo para la región de luma comprende: realizar submuestreo para la región vecina adyacente al bloque de luma basándose en:

$$pLeftDsY[0] = (pY[-3][0] + 2 * pY[-2][0] + pY[-1][0] + 2) >> 2.$$

60 5. El método de la reivindicación 1, en donde realizar submuestreo para la región de luma comprende: realizar submuestreo para la región vecina adyacente al bloque de luma basándose en:

$$65 \quad pLeftDsY[y] = (pY[-1][2 * y] + pY[-1][2 * y+1] + 2 * pY[-2][2 * y] + 2 * pY[-2][2 * y+1] + pY[-3][2 * y] + pY[-3][2 * y+1] + 4) >> 3.$$

6. El método de la reivindicación 1, en donde realizar submuestreo para la región de luma comprende:

cuando la región vecina pertenece a una misma CTU que el bloque de luma, realizar submuestreo para la región vecina adyacente al bloque de luma basándose en:

$$pTopDsY[x] = (pY[2*x][-3] + pY[2*x-1][-2] + 4*pY[2*x][-2] + pY[2*x+1][-2] + pY[2*x][-1] + 4) \gg 3.$$

7. El método de la reivindicación 1, en donde realizar submuestreo para la región de luma comprende: cuando la región vecina pertenece a una CTU diferente que el bloque de luma, realizar submuestreo para la región vecina adyacente al bloque de luma basándose en:

$$pTopDsY[x] = (pY[2*x-1][-1] + 2*pY[2*x][-1] + pY[2*x+1][-1] + 2) \gg 2.$$

8. El método de la reivindicación 1, en donde realizar submuestreo para la región de luma comprende: cuando la región vecina pertenece a una misma CTU que el bloque de luma, realizar submuestreo para la región vecina adyacente al bloque de luma basándose en:

$$pTopDsY[x] = (pY[2*x-1][-2] + pY[2*x-1][-1] + 2*pY[2*x][-2] + 2*pY[2*x][-1] + pY[2*x+1][-2] + pY[2*x+1][-1] + 4) \gg 3.$$

9. Un método de codificación de una imagen, que comprende:

determinar (S300) un modo de intrapredicción de un bloque actual (400, 410);  
 realizar (S310) intrapredicción en el bloque actual (400, 410) basándose en el modo de intrapredicción,  
 en donde el bloque actual (400, 410) comprende un bloque de luma y un bloque de croma,  
 en donde el modo de intrapredicción comprende un modo de intrapredicción de luma para la intrapredicción del bloque de luma y un modo de intrapredicción de croma para la intrapredicción del bloque de croma; y  
 codificar un índice de MPM en un flujo de bits,  
 en donde el índice de MPM se deriva basándose en el modo de intrapredicción de luma y en una lista de MPM, y  
 en donde la lista de MPM comprende al menos uno del modo A, el modo A+n o el modo A-n, en donde el modo A representa un modo de intrapredicción de un bloque vecino,  
 en donde realizar la intrapredicción comprende:

especificar (S500) una región de luma para la referencia de intercomponentes del bloque de croma;  
 realizar (S510) un submuestreo para la región de luma;  
 derivar (S520) un parámetro para la referencia de intercomponentes del bloque de croma;  
 y  
 predecir (S530) el bloque de croma basándose en la región de luma submuestreada y el parámetro,

en donde la región de luma comprende un bloque de luma y una región vecina adyacente al bloque de luma, y realizar submuestreo para la región de luma comprende:

realizar submuestreo para el bloque de luma basándose en una de:

$$pDsY[x][y] = (pY[2 * x][2 * y-1] + pY[2 * x-1][2 * y] + 4 * pY[2 * x][2 * y] + pY[2 * x+1][2 * y] + pY[2 * x][2 * y+1] + 4) \gg 3, \text{ o}$$

o

$$pDsY[x][y] = (pY[2 * x-1][2 * y] + pY[2 * x-1][2 * y+1] + 2 * pY[2 * x][2 * y] + 2 * pY[2 * x][2 * y+1] + pY[2 * x+1][2 * y] + pY[2 * x+1][2 * y+1] + 4) \gg 3$$

en donde  $pY[a][b]$  representa un valor de píxel de un píxel que tiene las coordenadas (a, b).

10. Un decodificador, que comprende una memoria y un procesador, en donde

la memoria está configurada para almacenar programas informáticos que pueden ponerse en marcha en el procesador; y  
 el procesador está configurado para poner en marcha los programas informáticos para ejecutar el método de cualquiera de las reivindicaciones 1-8.

11. Un codificador, que comprende una memoria y un procesador, en donde

la memoria está configurada para almacenar programas informáticos que pueden ponerse en marcha en el primer procesador; y el procesador está configurado para poner en marcha los programas informáticos para ejecutar el método de la reivindicación 9.

- 5
12. Un medio de almacenamiento informático, que almacena programas informáticos, en donde los programas informáticos son ejecutados por un primer procesador para implementar el método de una cualquiera de las reivindicaciones 1-8, o son ejecutados por un segundo procesador para implementar el método de la reivindicación 9.
- 10

15

20

25

30

35

40

45

50

55

60

65

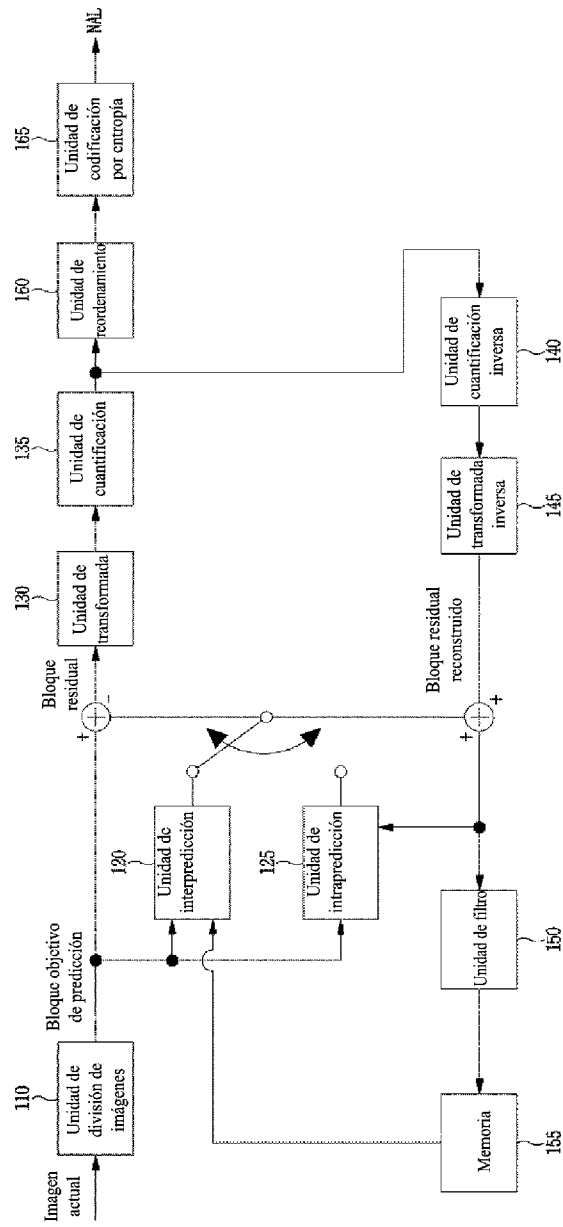


Figura 1

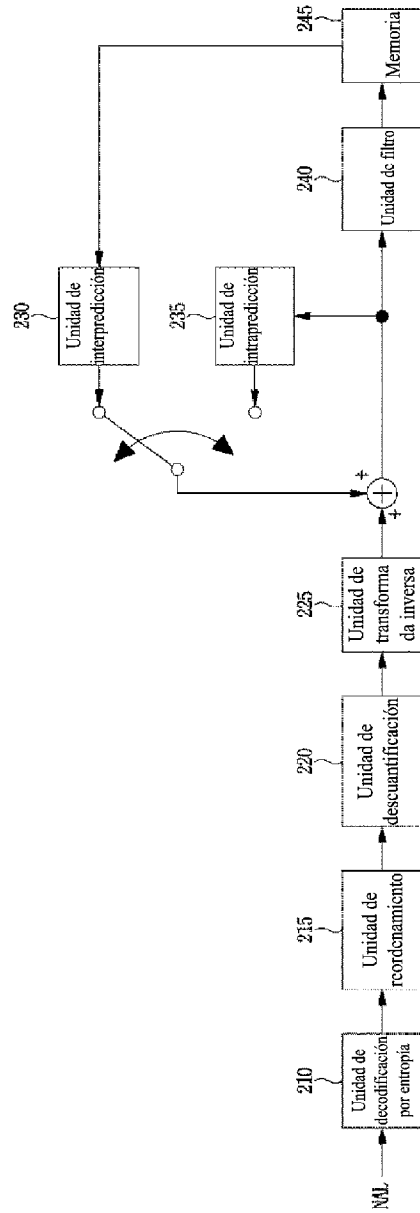


Figura 2

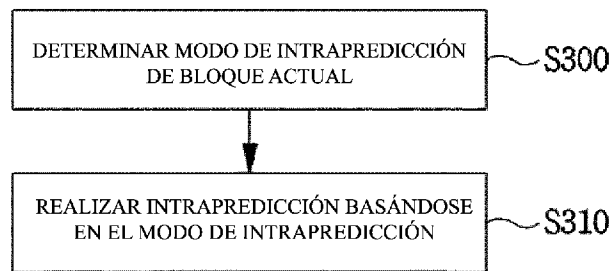


Figura 3

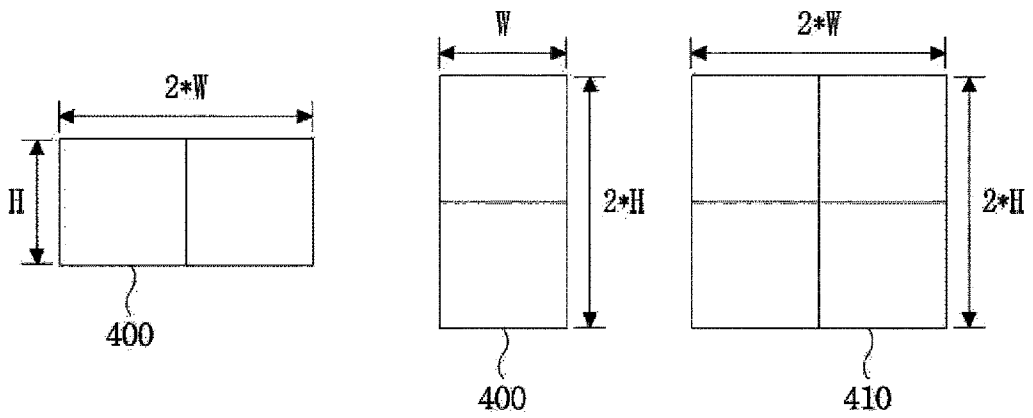
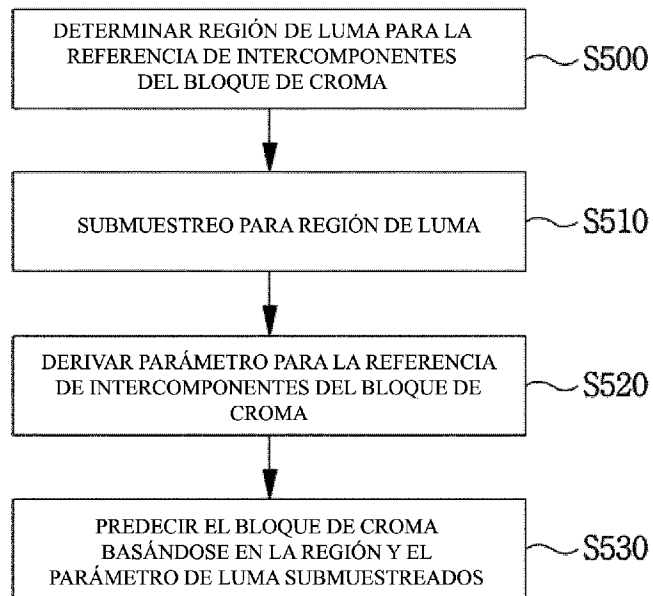


Figura 4



**Figura 5**