

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 985 144**

51 Int. Cl.:

H04N 19/157 (2014.01)

H04N 19/70 (2014.01)

H04N 19/186 (2014.01)

H04N 19/119 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **21.07.2020 PCT/KR2020/009618**

87 Fecha y número de publicación internacional: **28.01.2021 WO21015537**

96 Fecha de presentación y número de la solicitud europea: **21.07.2020 E 20843702 (0)**

97 Fecha y número de publicación de la concesión europea: **27.03.2024 EP 4002843**

54 Título: **Método de codificación/decodificación de imagen y dispositivo para señalar información de predicción de la componente de crominancia según que el modo de paleta sea aplicable y método para transmitir un tren de bits**

30 Prioridad:
21.07.2019 US 2019628766 P

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
04.11.2024

73 Titular/es:
**LG ELECTRONICS INC. (100.0%)
128, Yeoui-daero, Yeongdeungpo-gu
SEOUL 07336, KR**

72 Inventor/es:
JANG, HYEONG MOON

74 Agente/Representante:
ELZABURU, S.L.P

ES 2 985 144 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 de codificación/decodificación de imagen y dispositivo para señalar información de predicción de la componente de crominancia según que el modo de paleta sea aplicable y método para transmitir un tren de bits

Campo técnico

- 5 La presente divulgación se refiere a un método y aparato de codificación/decodificación de imagen y, más particularmente, a un método de codificación/decodificación de imagen para señalar información de predicción de la componente de crominancia dependiendo de si se aplica o no un modo de paleta, y a un método para transmitir un tren de bits generado por el método/aparato de codificación de imagen de la presente divulgación.

Antecedentes de la técnica

- 10 Recientemente, la demanda de imágenes de alta resolución y alta calidad, como imágenes de alta definición (HD) e imágenes de ultra alta definición (UHD), está aumentando en varios campos. A medida que se mejoran la resolución y la calidad de los datos de imagen, la cantidad de información o bits transmitidos aumenta relativamente en comparación con los datos de imagen existentes. Un aumento en la cantidad de información o bits transmitidos provoca un aumento en el costo de transmisión y el costo de almacenamiento.
- 15 En consecuencia, existe la necesidad de una tecnología de compresión de imagen de alta eficiencia para transmitir, almacenar y reproducir información de manera eficaz en imágenes de alta resolución y alta calidad. En la presentación "Versatile Video Coding (Draft 6)" para la 15ª reunión del JVET del 3 de julio de 2019 al 12 de julio de 2019 en GOTEMBURGO del EQUIPO CONJUNTO DE EXPLORACIÓN DE VIDEO DE ISO/IEC JTC1/SC29/WG11 Y UIT-T SG.16, publicada como documento JVET-O2001, BROSS B et al. divulga un método de decodificación de imágenes.
- 20 En la presentación "CE8-2.3: Compound Palette Mode" para la 15ª reunión del JVET del 3 de julio de 2019 al 12 de julio de 2019 en GOTEMBURGO del EQUIPO CONJUNTO DE EXPLORACIÓN DE VIDEO DE ISO/IEC JTC1/SC29/WG11 Y UIT-T SG.16, publicado como documento JVET-O0071, ZHU (BYTEDANCE), W et al. divulga un modo de paleta.

Divulgación

- 25 Problema técnico

Un objeto de la presente divulgación es proporcionar un método y aparato de codificación/decodificación de imagen con una eficiencia de codificación/decodificación mejorada.

- 30 Un objeto de la presente divulgación es proporcionar un método y aparato de codificación/decodificación de imagen para mejorar la eficiencia de codificación/decodificación mediante la señalización de información de predicción de la componente de crominancia dependiendo de si se aplica o no un modo de paleta.

Otro objeto de la presente divulgación es proporcionar un método para transmitir un tren de bits generado por un método o aparato de codificación de imagen según la presente divulgación.

Otro objeto de la presente divulgación es proporcionar un soporte de grabación que almacene un tren de bits generado por un método o aparato de codificación de imagen según la presente divulgación.

- 35 Otro objeto de la presente divulgación es proporcionar un soporte de grabación que almacene un tren de bits recibido, decodificado y utilizado para reconstruir una imagen mediante un aparato de decodificación de imagen según la presente divulgación.

Efectos ventajosos

- 40 Según la presente divulgación, es posible proporcionar un método y aparato de codificación/decodificación de imagen con una eficiencia de codificación/decodificación mejorada.

Además, según la presente divulgación, es posible proporcionar un método y aparato de codificación/decodificación de imagen para mejorar la eficiencia de codificación/decodificación mediante la señalización de información de predicción de la componente de crominancia dependiendo de si se aplica o no un modo de paleta.

- 45 Además, según la presente divulgación, es posible proporcionar un método para transmitir un tren de bits generado por un método o aparato de codificación de imagen según la presente divulgación.

Además, según la presente divulgación, es posible proporcionar un soporte de grabación que almacene un tren de bits generado por un método o aparato de codificación de imagen según la presente divulgación.

- 50 Además, según la presente divulgación, es posible proporcionar un soporte de grabación que almacene un tren de bits recibido, decodificado y utilizado para reconstruir una imagen mediante un aparato de decodificación de imagen según la presente divulgación.

Los expertos en la técnica apreciarán que los efectos que se pueden lograr a través de la presente divulgación no se limitan a lo que se ha descrito particularmente anteriormente en el presente documento y otras ventajas de la presente divulgación se entenderán más claramente a partir de la descripción detallada.

Descripción de dibujos

- 5 La FIG. 1 es una vista que muestra esquemáticamente un sistema de codificación de vídeo, al que es aplicable una realización de la presente divulgación.
- La FIG. 2 es una vista que muestra esquemáticamente un aparato de codificación de imagen, al que es aplicable una realización de la presente divulgación.
- 10 La FIG. 3 es una vista que muestra esquemáticamente un aparato de decodificación de imagen, al que es aplicable una realización de la presente divulgación.
- La FIG. 4 es una vista que muestra una estructura de particionado de una imagen según una realización.
- La FIG. 5 es una vista que muestra una realización de un tipo de particionado de un bloque según una estructura en árbol de tipos múltiples.
- 15 La FIG. 6 es una vista que muestra un mecanismo de señalización de información de división de bloques en un árbol cuádruple con estructura en árbol de tipos múltiples anidado según la presente divulgación.
- La FIG. 7 es una vista que muestra una realización en la que una CTU está subdividida en múltiples CU.
- La FIG. 8 es una vista que ilustra una realización de un patrón de división redundante.
- La FIG. 9 es una vista que ilustra una sintaxis para la señalización de formato de crominancia según una realización.
- 20 La FIG. 10 es una vista que ilustra una tabla de clasificación de formatos de crominancia según una realización.
- La FIG. 11 es una vista que ilustra la exploración horizontal y la exploración vertical según una realización.
- Las FIGS. 12 a 13 son vistas que ilustran una sintaxis para un modo de paleta según una realización.
- Las FIGS. 14 a 19 son vistas que ilustran una sintaxis para un modo de paleta según una realización.
- 25 La FIG. 20 es una vista que ilustra una ecuación para determinar PredictorPaletteEntries y CurrentPaletteEntries según una realización.
- La FIG. 21 es una vista que ilustra la sintaxis de una unidad de codificación según una realización modificada.
- La FIG. 22 es un diagrama de flujo que ilustra un método de señalización de información de predicción intra de crominancia predeterminada según una realización.
- 30 La FIG. 23 es un diagrama de flujo que ilustra un método para obtener información de predicción de crominancia mediante un aparato de decodificación según una realización.
- La FIG. 24 es un diagrama de flujo que ilustra un método para codificar una imagen mediante un aparato de codificación según una realización.
- La FIG. 25 es un diagrama de flujo que ilustra un método para decodificar una imagen mediante un aparato de decodificación según una realización.
- 35 La FIG. 26 es una vista que muestra un sistema de transmisión en flujo continuo, al que es aplicable una realización de la presente divulgación.

Modo de invención

40 A continuación, las realizaciones de la presente divulgación se describirán en detalle con referencia a los dibujos adjuntos para que los expertos en la técnica puedan implementarlas fácilmente. Sin embargo, la presente divulgación puede implementarse de varias formas diferentes; la invención se define en el conjunto de reivindicaciones adjuntas.

Al describir la presente divulgación, si se determina que la descripción detallada de una función o construcción conocida relacionada hace que el alcance de la presente divulgación sea innecesariamente ambiguo, se omitirá la descripción detallada de la misma. En los dibujos, se omiten partes no relacionadas con la descripción de la presente divulgación y se agregan números de referencia similares a partes similares.

45 En la presente divulgación, cuando un componente está "conectado", "acoplado" o "vinculado" a otro componente,

puede incluir no solo una relación de conexión directa, sino también una relación de conexión indirecta en la que un componente interviniente está presente. Además, cuando un componente "incluye" o "tiene" otros componentes, significa que se pueden incluir además otros componentes, en lugar de excluir otros componentes, a menos que se indique lo contrario.

5 En la presente divulgación, los términos primero, segundo, etc. pueden usarse solo con el fin de distinguir un componente de otros componentes, y no limitan el orden o la importancia de los componentes a menos que se indique lo contrario. Por consiguiente, dentro del alcance de la presente divulgación, un primer componente en una realización puede denominarse segundo componente en otra realización y, de manera similar, un segundo componente en una realización puede denominarse primer componente en otra realización.

10 En la presente divulgación, los componentes que se distinguen entre sí tienen como objetivo describir claramente cada característica y no significa que los componentes estén necesariamente separados. Es decir, se pueden integrar e implementar una pluralidad de componentes en una unidad de hardware o software, o un componente se puede distribuir e implementar en una pluralidad de unidades de hardware o software. Por lo tanto, incluso si no se indica lo contrario, aquellas realizaciones en las que los componentes están integrados o el componente está distribuido también están incluidas en el alcance de la presente divulgación.

15 En la presente divulgación, los componentes descritos en diversas realizaciones no significan necesariamente componentes esenciales y algunos componentes pueden ser componentes opcionales. Por consiguiente, también se incluye en el alcance de la presente divulgación una realización que consta de un subconjunto de componentes descritos en una realización. Además, las realizaciones que incluyen otros componentes además de los componentes descritos en las diversas realizaciones están incluidas en el alcance de la presente divulgación.

20 La presente divulgación se refiere a la codificación y decodificación de una imagen, y los términos utilizados en la presente divulgación pueden tener un significado general comúnmente usado en el campo técnico al que pertenece la presente divulgación, a menos que se definan nuevamente en la presente divulgación.

25 En la presente divulgación, un "fotograma" (*picture*) generalmente se refiere a una unidad que representa una imagen en un período de tiempo específico, y una porción/mosaico es una unidad de codificación que constituye una parte de un fotograma, y un fotograma puede estar compuesto por una o más porciones/mosaicos. Además, una porción/mosaico puede incluir una o más unidades de árbol de codificación (CTU).

30 En la presente divulgación, un "píxel" o un "pel" puede significar una unidad más pequeña que constituye un fotograma (o imagen). Además, se puede utilizar "muestra" como término correspondiente a un píxel. Una muestra generalmente puede representar un píxel o un valor de un píxel, y puede representar solo un píxel/valor de píxel de una componente de luminancia o solo un píxel/valor de píxel de una componente de crominancia.

35 En la presente divulgación, una "unidad" puede representar una unidad básica de procesamiento de imagen. La unidad puede incluir al menos una de una región específica del fotograma e información relacionada con la región. La unidad puede usarse de forma intercambiable con términos como "matriz de muestras", "bloque" o "área" en algunos casos. En un caso general, un bloque de MxN puede incluir muestras (o matrices de muestras) o un conjunto (o matriz) de coeficientes de la transformada de M columnas y N filas.

40 En la presente divulgación, "bloque actual" puede significar uno de "bloque de codificación actual", "unidad de codificación actual", "bloque objetivo de codificación", "bloque objetivo de decodificación" o "bloque objetivo de procesamiento". Cuando se realiza la predicción, "bloque actual" puede significar "bloque de predicción actual" o "bloque objetivo de predicción". Cuando se realiza la transformada (transformada inversa)/cuantificación (descuantificación), "bloque actual" puede significar "bloque de transformada actual" o "bloque objetivo de transformada". Cuando se realiza el filtrado, "bloque actual" puede significar "bloque objetivo de filtrado".

45 Además, en la presente divulgación, un "bloque actual" puede significar "un bloque de luminancia de un bloque actual" a menos que se indique explícitamente como un bloque de crominancia. El "bloque de crominancia del bloque actual" puede expresarse incluyendo una descripción explícita de un bloque de crominancia, tal como "bloque de crominancia" o "bloque de crominancia actual".

En la presente divulgación, los términos "/" y "," deben interpretarse en el sentido de que indican "y/o". Por ejemplo, la expresión "A/B" y "A, B" puede significar "A y/o B". Además, "A/B/C" y "A/B/C" pueden significar "al menos uno de A, B y/o C".

50 En la presente divulgación, el término "o" debe interpretarse en el sentido de que indica "y/o". Por ejemplo, la expresión "A o B" puede comprender 1) solo "A", 2) solo "B" y/o 3) "tanto A como B". En otras palabras, en la presente divulgación, el término "o" debe interpretarse en el sentido de que indica "adicional o alternativamente".

Descripción general del sistema de codificación de vídeo

La FIG. 1 es una vista que muestra un sistema de codificación de vídeo según la presente divulgación.

El sistema de codificación de vídeo según una realización puede incluir un aparato de codificación 10 y un aparato de decodificación 20. El aparato de codificación 10 puede entregar información o datos de vídeo y/o imagen codificados al aparato de decodificación 20 en forma de un archivo o flujo continuo a través de un soporte o red de almacenamiento digital.

5 El aparato de codificación 10 según una realización puede incluir un generador de fuente de vídeo 11, una unidad de codificación 12 y un transmisor 13. El aparato de decodificación 20 según una realización puede incluir un receptor 21, una unidad de decodificación 22 y un renderizador 23. La unidad de codificación 12 puede denominarse unidad de codificación de vídeo/imagen, y la unidad de decodificación 22 puede denominarse unidad de decodificación de vídeo/imagen. El transmisor 13 puede incluirse en la unidad de codificación 12. El receptor 21 puede incluirse en la
10 unidad de decodificación 22. El renderizador 23 puede incluir una pantalla y la pantalla puede configurarse como un dispositivo separado o un componente externo.

El generador de fuente de vídeo 11 puede adquirir un vídeo/imagen a través de un proceso de captura, sintetización o generación del vídeo/imagen. El generador de fuente de vídeo 11 puede incluir un dispositivo de captura de vídeo/imagen y/o un dispositivo de generación de vídeo/imagen. El dispositivo de captura de vídeo/imagen puede
15 incluir, por ejemplo, una o más cámaras, archivos de vídeo/imagen que incluyen vídeos/imágenes capturados previamente y similares. El dispositivo de generación de vídeo/imagen puede incluir, por ejemplo, computadoras, tabletas y teléfonos inteligentes, y puede generar (electrónicamente) vídeo/imágenes. Por ejemplo, se puede generar un vídeo/imagen virtual a través de una computadora o similar. En este caso, el proceso de captura de vídeo/imagen puede sustituirse por un proceso de generación de datos relacionados.

20 La unidad de codificación 12 puede codificar un vídeo/imagen de entrada. La unidad de codificación 12 puede realizar una serie de procedimientos tales como predicción, transformada y cuantificación para la eficiencia de compresión y codificación. La unidad de codificación 12 puede emitir datos codificados (información de vídeo/imagen codificada) en forma de un tren de bits.

El transmisor 13 puede transmitir la información de vídeo/imagen codificada o la salida de datos en forma de un tren de bits al receptor 21 del aparato de decodificación 20 a través de un soporte de almacenamiento digital o una red en
25 forma de un archivo o flujo continuo. El soporte de almacenamiento digital puede incluir varios soportes de almacenamiento tales como USB, SD, CD, DVD, Blu-ray, HDD, SSD y similares. El transmisor 13 puede incluir un elemento para generar un archivo multimedia a través de un formato de archivo predeterminado y puede incluir un elemento para la transmisión a través de una red de difusión/comunicación. El receptor 21 puede extraer/recibir el tren de bits del soporte o red de almacenamiento y transmitir el tren de bits a la unidad de decodificación 22.
30

La unidad de decodificación 22 puede decodificar el vídeo/imagen realizando una serie de procedimientos tales como descuantificación, transformada inversa y predicción correspondientes al funcionamiento de la unidad de codificación 12.

El renderizador 23 puede renderizar el vídeo/imagen decodificado. El vídeo/imagen renderizado se puede presentar a través de la pantalla.

35 Descripción general del aparato de codificación de imagen

La FIG. 2 es una vista que muestra esquemáticamente un aparato de codificación de imagen, al que es aplicable una realización de la presente divulgación.

Como se muestra en la FIG. 2, el aparato de codificación de imagen 100 puede incluir un particionador de imagen 110, un restador 115, un transformador 120, un cuantificador 130, un descuantificador 140, un transformador inverso
40 150, un sumador 155, un filtro 160, una memoria 170, una unidad de predicción inter 180, una unidad de predicción intra 185 y un codificador de entropía 190. La unidad de predicción inter 180 y la unidad de predicción intra 185 pueden denominarse colectivamente "unidad de predicción". El transformador 120, el cuantificador 130, el descuantificador 140 y el transformador inverso 150 pueden incluirse en un procesador residual. El procesador residual puede incluir además el restador 115.

45 Todos o al menos algunos de la pluralidad de componentes que configuran el aparato de codificación de imagen 100 pueden configurarse mediante un componente de hardware (por ejemplo, un codificador o un procesador) en algunas realizaciones. Además, la memoria 170 puede incluir una memoria intermedia de fotogramas decodificados (DPB) y puede configurarse mediante un soporte de almacenamiento digital.

El particionador de imagen 110 puede subdividir una imagen de entrada (o un fotograma o un cuadro) introducida en el aparato de codificación de imagen 100 en una o más unidades de procesamiento. Por ejemplo, la unidad de procesamiento puede denominarse unidad de codificación (CU). La unidad de codificación puede adquirirse subdividiendo recurrentemente una unidad de árbol de codificación (CTU) o una unidad de codificación más grande (LCU) según una estructura en árbol cuádruple, en árbol binario, en árbol ternario (QT/BT/TT). Por ejemplo, una unidad de codificación puede subdividirse en una pluralidad de unidades de codificación de una mayor profundidad basándose
50 en una estructura en árbol cuádruple, una estructura en árbol binario y/o una estructura ternaria. Para subdividir la unidad de codificación, se puede aplicar primero una estructura en árbol cuádruple y después se puede aplicar la estructura en árbol binario y/o la estructura ternaria. El procedimiento de codificación según la presente divulgación se
55

puede realizar basándose en la unidad de codificación final que ya no se subdivide. La unidad de codificación más grande puede usarse como unidad de codificación final o la unidad de codificación de mayor profundidad adquirida al subdividir la unidad de codificación más grande puede usarse como unidad de codificación final. Aquí, el procedimiento de codificación puede incluir un procedimiento de predicción, transformada y reconstrucción, que se describirá más adelante. Como otro ejemplo, la unidad de procesamiento del procedimiento de codificación puede ser una unidad de predicción (PU) o una unidad de transformada (TU). La unidad de predicción y la unidad de transformada pueden dividirse o subdividirse a partir de la unidad de codificación final. La unidad de predicción puede ser una unidad de predicción de muestras, y la unidad de transformada puede ser una unidad para derivar un coeficiente de la transformada y/o una unidad para derivar una señal residual a partir del coeficiente de la transformada.

5
10
15

La unidad de predicción (la unidad de predicción inter 180 o la unidad de predicción intra 185) puede realizar predicción en un bloque que va a procesarse (bloque actual) y generar un bloque predicho que incluye muestras de predicción para el bloque actual. La unidad de predicción puede determinar si la predicción intra o la predicción inter se aplica por bloque actual o por CU. La unidad de predicción puede generar diversa información relacionada con la predicción del bloque actual y transmitir la información generada al codificador de entropía 190. La información sobre la predicción puede codificarse en el codificador de entropía 190 y emitirse en forma de un tren de bits.

20

La unidad de predicción intra 185 puede predecir el bloque actual haciendo referencia a las muestras en el fotograma actual. Las muestras referidas pueden estar ubicadas en la vecindad del bloque actual o pueden estar ubicadas separadas según el modo de predicción intra y/o la técnica de predicción intra. Los modos de predicción intra pueden incluir una pluralidad de modos no direccionales y una pluralidad de modos direccionales. El modo no direccional puede incluir, por ejemplo, un modo DC y un modo plano. El modo direccional puede incluir, por ejemplo, 33 modos de predicción direccional o 65 modos de predicción direccional según el grado de detalle de la dirección de predicción. Sin embargo, esto es simplemente un ejemplo; se pueden usar más o menos modos de predicción direccional dependiendo de la configuración. La unidad de predicción intra 185 puede determinar el modo de predicción aplicado al bloque actual utilizando un modo de predicción aplicado a un bloque vecino.

25
30
35
40
45

La unidad de predicción inter 180 puede derivar un bloque predicho para el bloque actual basándose en un bloque de referencia (matriz de muestras de referencia) especificado por un vector de movimiento en un fotograma de referencia. En este caso, para reducir la cantidad de información de movimiento transmitida en el modo de predicción inter, la información de movimiento se puede predecir en unidades de bloques, subbloques o muestras basándose en la correlación de información de movimiento entre el bloque vecino y el bloque actual. La información de movimiento puede incluir un vector de movimiento y un índice de fotograma de referencia. La información de movimiento puede incluir además información de dirección de predicción inter (predicción L0, predicción L1, predicción Bi, etc.). En el caso de predicción inter, el bloque vecino puede incluir un bloque vecino espacial presente en el fotograma actual y un bloque vecino temporal presente en el fotograma de referencia. El fotograma de referencia que incluye el bloque de referencia y el fotograma de referencia que incluye el bloque vecino temporal pueden ser iguales o diferentes. El bloque vecino temporal puede denominarse bloque de referencia coubicado, CU coubicada (colCU) y similares. El fotograma de referencia que incluye el bloque vecino temporal puede denominarse fotograma coubicado (colPic). Por ejemplo, la unidad de predicción inter 180 puede configurar una lista de candidatos de información de movimiento basada en bloques vecinos y generar información que especifica qué candidato se usa para derivar un vector de movimiento y/o un índice de fotograma de referencia del bloque actual. La predicción inter se puede realizar basándose en varios modos de predicción. Por ejemplo, en el caso de un modo de salto y un modo de fusión, la unidad de predicción inter 180 puede usar información de movimiento del bloque vecino como información de movimiento del bloque actual. En el caso del modo de salto, a diferencia del modo de fusión, es posible que la señal residual no se transmita. En el caso del modo de predicción del vector de movimiento (MVP), el vector de movimiento del bloque vecino se puede utilizar como predictor del vector de movimiento, y el vector de movimiento del bloque actual se puede señalar codificando una diferencia del vector de movimiento y un indicador de un predictor de vector de movimiento. La diferencia del vector de movimiento puede significar una diferencia entre el vector de movimiento del bloque actual y el predictor del vector de movimiento.

50
55
60

La unidad de predicción puede generar una señal de predicción basada en varios métodos de predicción y técnicas de predicción que se describen a continuación. Por ejemplo, la unidad de predicción puede no solo aplicar predicción intra o predicción inter, sino también aplicar simultáneamente tanto predicción intra como predicción inter, para predecir el bloque actual. Un método de predicción que aplica simultáneamente tanto la predicción intra como la predicción inter para la predicción del bloque actual puede denominarse predicción inter e intra combinada (CIIP). Además, la unidad de predicción puede realizar una copia de bloque intra (IBC) para la predicción del bloque actual. La copia de bloque intra puede usarse para codificación de imagen/vídeo de contenido de un juego o similar, por ejemplo, codificación de contenido de pantalla (SCC). IBC es un método para predecir un fotograma actual utilizando un bloque de referencia previamente reconstruido en el fotograma actual en una ubicación separada del bloque actual una distancia predeterminada. Cuando se aplica IBC, la ubicación del bloque de referencia en el fotograma actual puede codificarse como un vector (vector de bloque) correspondiente a la distancia predeterminada. Básicamente, IBC realiza una predicción en el fotograma actual, pero se puede realizar de manera similar a la predicción inter en el sentido de que se deriva un bloque de referencia dentro del fotograma actual. Es decir, IBC puede usar al menos una de las técnicas de predicción inter descritas en la presente divulgación.

La señal de predicción generada por la unidad de predicción puede usarse para generar una señal reconstruida o para

generar una señal residual. El restador 115 puede generar una señal residual (bloque residual o matriz de muestras residuales) restando la señal de predicción (bloque predicho o matriz de muestras de predicción) emitida desde la unidad de predicción de la señal de imagen de entrada (bloque original o matriz de muestras original). La señal residual generada puede transmitirse al transformador 120.

- 5 El transformador 120 puede generar coeficientes de transformada aplicando una técnica de transformada a la señal residual. Por ejemplo, la técnica de transformada puede incluir al menos una de una transformada de coseno discreta (DCT), una transformada de seno discreta (DST), una transformada de karhunen-loève (KLT), una transformada basada en gráficos (GBT) o una transformada condicionalmente no lineal (CNT). Aquí, GBT significa transformada obtenida de un gráfico cuando el gráfico representa la información de relación entre píxeles. La CNT se refiere a la transformada adquirida basándose en una señal de predicción generada utilizando todos los píxeles previamente reconstruidos. Además, el proceso de transformada puede aplicarse a bloques de píxeles cuadrados que tengan el mismo tamaño o puede aplicarse a bloques que tengan un tamaño variable en lugar de cuadrados.

- 10 El cuantificador 130 puede cuantificar los coeficientes de transformada y transmitirlos al codificador de entropía 190. El codificador de entropía 190 puede codificar la señal cuantificada (información sobre los coeficientes de transformada cuantificados) y emitir un tren de bits. La información sobre los coeficientes de transformada cuantificados puede denominarse información residual. El cuantificador 130 puede reorganizar los coeficientes de transformada cuantificados en forma de bloque en una forma vectorial unidimensional basándose en un orden de exploración de coeficientes y generar información sobre los coeficientes de transformada cuantificados basándose en los coeficientes de transformada cuantificados en la forma vectorial unidimensional.

- 15 El codificador de entropía 190 puede realizar diversos métodos de codificación tales como, por ejemplo, Golomb exponencial, codificación de longitud variable adaptativa al contexto (CAVLC), codificación aritmética binaria adaptativa al contexto (CABAC) y similares. El codificador de entropía 190 puede codificar información necesaria para la reconstrucción de vídeo/imagen distinta de los coeficientes de transformada cuantificados (por ejemplo, valores de elementos de sintaxis, etc.) juntos o por separado. La información codificada (por ejemplo, información de vídeo/imagen codificada) puede transmitirse o almacenarse en unidades de capas de abstracción de red (NAL) en forma de un tren de bits. La información de vídeo/imagen puede incluir además información sobre diversos conjuntos de parámetros tales como un conjunto de parámetros de adaptación (APS), un conjunto de parámetros de fotograma (PPS), un conjunto de parámetros de secuencia (SPS) o un conjunto de parámetros de vídeo (VPS). Además, la información de vídeo/imagen puede incluir además información de restricción general. La información señalizada, la información transmitida y/o los elementos de sintaxis descritos en la presente divulgación pueden codificarse mediante el procedimiento de codificación descrito anteriormente e incluirse en el tren de bits.

- 20 El tren de bits puede transmitirse a través de una red o puede almacenarse en un soporte de almacenamiento digital. La red puede incluir una red de difusión y/o una red de comunicación, y el soporte de almacenamiento digital puede incluir varios soportes de almacenamiento tales como USB, SD, CD, DVD, Blu-ray, HDD, SSD y similares. Un transmisor (no mostrado) que transmite una señal emitida desde el codificador de entropía 190 y/o una unidad de almacenamiento (no mostrada) que almacena la señal puede incluirse como elemento interno/externo del aparato de codificación de imagen 100. Alternativamente, el transmisor puede proporcionarse como componente del codificador de entropía 190.

- 25 Los coeficientes de transformada cuantificados emitidos desde el cuantificador 130 pueden usarse para generar una señal residual. Por ejemplo, la señal residual (bloque residual o muestras residuales) puede reconstruirse aplicando descuantificación y transformada inversa a los coeficientes de transformada cuantificados a través del descuantificador 140 y el transformador inverso 150.

- 30 El sumador 155 añade la señal residual reconstruida a la señal de predicción emitida desde la unidad de predicción inter 180 o la unidad de predicción intra 185 para generar una señal reconstruida (fotograma reconstruido, bloque reconstruido, matriz de muestras reconstruida). Si no hay residuo para el bloque que va a procesarse, como en el caso en el que se aplica el modo de salto, el bloque predicho puede usarse como bloque reconstruido. El sumador 155 puede denominarse reconstructor o generador de bloques reconstruidos. La señal reconstruida generada se puede usar para la predicción intra de un siguiente bloque que vaya a procesarse en el fotograma actual y se puede usar para la predicción inter de un fotograma siguiente mediante filtrado como se describe a continuación.

- 35 El filtro 160 puede mejorar la calidad de imagen subjetiva/objetiva aplicando filtrado a la señal reconstruida. Por ejemplo, el filtro 160 puede generar un fotograma reconstruido modificado aplicando varios métodos de filtrado al fotograma reconstruido y almacenar el fotograma reconstruido modificado en la memoria 170, específicamente, un DPB de la memoria 170. Los diversos métodos de filtrado pueden incluir, por ejemplo, filtrado de desbloqueo, compensación adaptativa de muestras, filtro de bucle adaptativo, un filtro bilateral, etc. El filtro 160 puede generar diversa información relacionada con el filtrado y transmitir la información generada al codificador de entropía 190 como se describe más adelante en la descripción de cada método de filtrado. La información relacionada con el filtrado puede codificarse mediante el codificador de entropía 190 y emitirse en forma de un tren de bits.

40 El fotograma reconstruido modificado transmitido a la memoria 170 puede usarse como fotograma de referencia en la unidad de predicción inter 180. Cuando se aplica predicción inter a través del aparato de codificación de imagen 100,

la discrepancia de predicción entre el aparato de codificación de imagen 100 y el aparato de decodificación de imagen puede evitarse y puede mejorarse mejorando la eficiencia de codificación.

El DPB de la memoria 170 puede almacenar el fotograma reconstruido modificado para su uso como fotograma de referencia en la unidad de predicción inter 180. La memoria 170 puede almacenar la información de movimiento del bloque del cual se deriva (o codifica) la información de movimiento en el fotograma actual y/o la información de movimiento de los bloques del fotograma que ya han sido reconstruidos. La información de movimiento almacenada puede transmitirse a la unidad de predicción inter 180 y usarse como información de movimiento del bloque vecino espacial o información de movimiento del bloque vecino temporal. La memoria 170 puede almacenar muestras reconstruidas de bloques reconstruidos en el fotograma actual y puede transferir las muestras reconstruidas a la unidad de predicción intra 185.

Descripción general del aparato de decodificación de imagen

La FIG. 3 es una vista que muestra esquemáticamente un aparato de decodificación de imagen, al que es aplicable una realización de la presente divulgación.

Como se muestra en la FIG. 3, el aparato de decodificación de imagen 200 puede incluir un decodificador de entropía 210, un descuantificador 220, un transformador inverso 230, un sumador 235, un filtro 240, una memoria 250, una unidad de predicción inter 260 y una unidad de predicción intra 265. La unidad de predicción inter 260 y la unidad de predicción intra 265 pueden denominarse colectivamente "unidad de predicción". El descuantificador 220 y el transformador inverso 230 pueden incluirse en un procesador residual.

Todos o al menos algunos de una pluralidad de componentes que configuran el aparato de decodificación de imagen 200 pueden configurarse mediante un componente de hardware (por ejemplo, un decodificador o un procesador) según una realización. Además, la memoria 250 puede incluir una memoria intermedia de fotogramas decodificados (DPB) o puede configurarse mediante un soporte de almacenamiento digital.

El aparato de decodificación de imagen 200, que ha recibido un tren de bits que incluye información de vídeo/imagen, puede reconstruir una imagen realizando un proceso correspondiente a un proceso realizado por el aparato de codificación de imagen 100 de la FIG. 2. Por ejemplo, el aparato de decodificación de imagen 200 puede realizar la decodificación usando una unidad de procesamiento aplicada en el aparato de codificación de imagen. Así, la unidad de procesamiento de decodificación puede ser, por ejemplo, una unidad de codificación. La unidad de codificación puede adquirirse subdividiendo una unidad de árbol de codificación o una unidad de codificación más grande. La señal de imagen reconstruida decodificada y emitida a través del aparato de decodificación de imagen 200 puede reproducirse a través de un aparato de reproducción (no mostrado).

El aparato de decodificación de imagen 200 puede recibir una señal emitida desde el aparato de codificación de imagen de la FIG. 2 en forma de tren de bits. La señal recibida puede decodificarse a través del decodificador de entropía 210. Por ejemplo, el decodificador de entropía 210 puede analizar el tren de bits para derivar información (por ejemplo, información de vídeo/imagen) necesaria para la reconstrucción de imagen (o reconstrucción de fotogramas). La información de vídeo/imagen puede incluir además información sobre diversos conjuntos de parámetros tales como un conjunto de parámetros de adaptación (APS), un conjunto de parámetros de fotograma (PPS), un conjunto de parámetros de secuencia (SPS) o un conjunto de parámetros de vídeo (VPS). Además, la información de vídeo/imagen puede incluir además información de restricción general. El aparato de decodificación de imagen puede decodificar además un fotograma basándose en la información sobre el conjunto de parámetros y/o la información de restricción general. La información señalizada/recibida y/o los elementos de sintaxis descritos en la presente divulgación pueden decodificarse mediante el procedimiento de decodificación y obtenerse del tren de bits. Por ejemplo, el decodificador de entropía 210 decodifica la información en el tren de bits basándose en un método de codificación tal como codificación exponencial de Golomb, CAVLC o CABAC, y emite valores de elementos de sintaxis necesarios para la reconstrucción de imagen y valores cuantificados de coeficientes de transformada para residuos. Más específicamente, el método de decodificación de entropía CABAC puede recibir un bin correspondiente a cada elemento de sintaxis en el tren de bits, determinar un modelo de contexto usando información de un elemento de sintaxis objetivo de decodificación, información de decodificación de un bloque vecino y un bloque objetivo de decodificación o información de un símbolo/bin decodificado en una etapa anterior, y realiza decodificación aritmética en el bin prediciendo una probabilidad de ocurrencia de un bin de acuerdo con el modelo de contexto determinado, y genera un símbolo correspondiente al valor de cada elemento de sintaxis. En este caso, el método de decodificación de entropía CABAC puede actualizar el modelo de contexto utilizando la información del símbolo/bin decodificado para un modelo de contexto de un siguiente símbolo/bin después de determinar el modelo de contexto. La información relacionada con la predicción entre la información decodificada por el decodificador de entropía 210 puede proporcionarse a la unidad de predicción (la unidad de predicción inter 260 y la unidad de predicción intra 265), y el valor residual sobre el que se realizó la decodificación de entropía en el decodificador de entropía 210, es decir, los coeficientes de transformada cuantificados y la información de parámetros relacionados, se pueden introducir en el descuantificador 220. Además, se puede proporcionar al filtro 240 información sobre el filtrado entre la información decodificada por el decodificador de entropía 210. Mientras tanto, un receptor (no mostrado) para recibir una salida de señal desde el aparato de codificación de imagen puede configurarse además como un elemento interno/externo del aparato de decodificación de imagen 200, o el receptor puede ser un componente del decodificador de entropía 210.

- 5 Mientras tanto, el aparato de decodificación de imagen según la presente divulgación puede denominarse aparato de decodificación de vídeo/imagen/fotograma. El aparato de decodificación de imagen puede clasificarse en un decodificador de información (decodificador de información de vídeo/imagen/fotograma) y un decodificador de muestras (decodificador de muestras de vídeo/imagen/fotograma). El decodificador de información puede incluir el decodificador de entropía 210. El decodificador de muestras puede incluir al menos uno del descuantificador 220, el transformador inverso 230, el sumador 235, el filtro 240, la memoria 250, la unidad de predicción inter 260 o la unidad de predicción intra 265.
- 10 El descuantificador 220 puede descuantificar los coeficientes de transformada cuantificados y emitir los coeficientes de transformada. El descuantificador 220 puede reorganizar los coeficientes de transformada cuantificados en forma de un bloque bidimensional. En este caso, la reorganización se puede realizar basándose en el orden de exploración de coeficientes realizado en el aparato de codificación de imagen. El descuantificador 220 puede realizar la descuantificación de los coeficientes de transformada cuantificados utilizando un parámetro de cuantificación (por ejemplo, información del tamaño del paso de cuantificación) y obtener coeficientes de transformada.
- 15 El transformador inverso 230 puede transformar inversamente los coeficientes de transformada para obtener una señal residual (bloque residual, matriz de muestras residuales).
- La unidad de predicción puede realizar una predicción en el bloque actual y generar un bloque predicho que incluye muestras de predicción para el bloque actual. La unidad de predicción puede determinar si se aplica predicción intra o predicción inter al bloque actual basándose en la información sobre la salida de predicción del decodificador de entropía 210 y puede determinar un modo de predicción intra/inter específico (técnica de predicción).
- 20 Es lo mismo que se describe en la unidad de predicción del aparato de codificación de imagen 100, que la unidad de predicción puede generar la señal de predicción basándose en varios métodos (técnicas) de predicción que se describirán más adelante.
- La unidad de predicción intra 265 puede predecir el bloque actual mediante referencia a las muestras en el fotograma actual. La descripción de la unidad de predicción intra 185 se aplica igualmente a la unidad de predicción intra 265.
- 25 La unidad de predicción inter 260 puede derivar un bloque predicho para el bloque actual basándose en un bloque de referencia (matriz de muestras de referencia) especificado por un vector de movimiento en un fotograma de referencia. En este caso, para reducir la cantidad de información de movimiento transmitida en el modo de predicción inter, la información de movimiento se puede predecir en unidades de bloques, subbloques o muestras basándose en la correlación de la información de movimiento entre el bloque vecino y el bloque actual. La información de movimiento puede incluir un vector de movimiento y un índice de fotograma de referencia. La información de movimiento puede incluir además información de dirección de predicción inter (predicción L0, predicción L1, predicción Bi, etc.). En el caso de predicción inter, el bloque vecino puede incluir un bloque vecino espacial presente en el fotograma actual y un bloque vecino temporal presente en el fotograma de referencia. Por ejemplo, la unidad de predicción inter 260 puede configurar una lista de candidatos de información de movimiento basándose en bloques vecinos y derivar un vector de movimiento del bloque actual y/o un índice de fotograma de referencia basándose en la información de selección de candidatos recibida. La predicción inter se puede realizar basándose en varios modos de predicción, y la información sobre la predicción puede incluir información que especifica un modo de predicción inter para el bloque actual.
- 30
- 35 El sumador 235 puede generar una señal reconstruida (fotograma reconstruido, bloque reconstruido, matriz de muestras reconstruida) sumando la señal residual obtenida a la señal de predicción (bloque predicho, matriz de muestras predicha) emitida desde la unidad de predicción (que incluye la unidad de predicción inter 260 y/o la unidad de predicción intra 265). Si no hay residuos para el bloque que va a procesarse, como cuando se aplica el modo de salto, el bloque predicho puede usarse como bloque reconstruido. La descripción del sumador 155 es igualmente aplicable al sumador 235. El sumador 235 puede denominarse reconstructor o generador de bloques reconstruidos. La señal reconstruida generada se puede usar para la predicción intra de un siguiente bloque que va a procesarse en el fotograma actual y se puede usar para la predicción inter de un siguiente fotograma mediante filtrado como se describe a continuación.
- 40
- 45 El filtro 240 puede mejorar la calidad de imagen subjetiva/objetiva aplicando filtrado a la señal reconstruida. Por ejemplo, el filtro 240 puede generar un fotograma reconstruido modificado aplicando varios métodos de filtrado al fotograma reconstruido y almacenar el fotograma reconstruido modificado en la memoria 250, específicamente, un DPB de la memoria 250. Los diversos métodos de filtrado pueden incluir, por ejemplo, filtrado de desbloqueo, una compensación adaptativa de muestras, un filtro de bucle adaptativo, un filtro bilateral y similares.
- 50
- 55 El fotograma reconstruido (modificado) almacenado en el DPB de la memoria 250 puede usarse como fotograma de referencia en la unidad de predicción inter 260. La memoria 250 puede almacenar la información de movimiento del bloque del cual se deriva (o decodifica) la información de movimiento en el fotograma actual y/o la información de movimiento de los bloques en el fotograma que ya han sido reconstruidos. La información de movimiento almacenada puede transmitirse a la unidad de predicción inter 260 para ser utilizada como información de movimiento del bloque vecino espacial o información de movimiento del bloque vecino temporal. La memoria 250 puede almacenar muestras reconstruidas de bloques reconstruidos en el fotograma actual y transferir las muestras reconstruidas a la unidad de

predicción intra 265.

En la presente divulgación, las realizaciones descritas en el filtro 160, la unidad de predicción inter 180 y la unidad de predicción intra 185 del aparato de codificación de imagen 100 pueden aplicarse de manera igual o correspondiente al filtro 240, la unidad de predicción inter 260 y la unidad de predicción intra 265 del aparato de decodificación de imagen 200.

Descripción general del particionado de imagen

El método de codificación de vídeo/imagen según la presente divulgación se puede realizar basándose en una estructura de particionado de imagen como sigue. Específicamente, los procedimientos de predicción, procesamiento residual (transformada (inversa), (des)cuantificación, etc.), codificación de elementos de sintaxis y filtrado, que se describirán más adelante, se pueden realizar basándose en una CTU, CU (y/o TU, PU) derivadas en función de la estructura de particionado de imagen. La imagen puede subdividirse en unidades de bloques y el procedimiento de particionado de bloques puede realizarse en el particionador de imagen 110 del aparato de codificación. La información relacionada con el particionado puede codificarse mediante el codificador de entropía 190 y transmitirse al aparato de decodificación en forma de un tren de bits. El decodificador de entropía 210 del aparato de decodificación puede derivar una estructura de particionado de bloques del fotograma actual basándose en la información relacionada con el particionado obtenida del tren de bits y, basándose en esta, puede realizar una serie de procedimientos (por ejemplo, predicción, procesamiento residual, reconstrucción de bloque/fotograma, filtrado en bucle, etc.) para la decodificación de imagen.

Los fotogramas pueden subdividirse en una secuencia de unidades de árbol de codificación (CTU). La FIG. 4 muestra un ejemplo en el que un fotograma está subdividido en CTU. La CTU puede corresponder a un bloque de árbol de codificación (CTB). Alternativamente, la CTU puede incluir un bloque de árbol de codificación de muestras de luminancia y dos bloques de árbol de codificación de muestras de crominancia correspondientes. Por ejemplo, para un fotograma que contiene tres matrices de muestras, la CTU puede incluir un bloque de $N \times N$ de muestras de luminancia y dos bloques correspondientes de muestras de crominancia.

Descripción general del particionado de CTU

Como se describió anteriormente, la unidad de codificación puede adquirirse subdividiendo recurrentemente la unidad de árbol de codificación (CTU) o la unidad de codificación más grande (LCU) según una estructura en árbol cuádruple/árbol binario/árbol ternario (QT/BT/TT). Por ejemplo, la CTU puede subdividirse primero en estructuras en árbol cuádruple. A partir de entonces, los nodos hoja de la estructura en árbol cuádruple pueden subdividirse aún más mediante una estructura en árbol de tipos múltiples.

Particionar según un árbol cuádruple significa que una CU (o CTU) actual se subdivide en cuatro partes iguales. Al subdividir según un árbol cuádruple, la CU actual se puede subdividir en cuatro CU que tengan la misma anchura y la misma altura. Cuando la CU actual ya no se subdivide en la estructura de árbol cuádruple, la CU actual corresponde al nodo hoja de la estructura en árbol cuádruple. La CU correspondiente al nodo hoja de la estructura en árbol cuádruple ya no puede subdividirse y puede usarse como la unidad de codificación final descrita anteriormente. Alternativamente, la CU correspondiente al nodo hoja de la estructura en árbol cuádruple puede subdividirse además mediante una estructura en árbol de tipos múltiples.

La FIG. 5 es una vista que muestra una realización de un tipo de particionado de un bloque según una estructura en árbol de tipos múltiples. El particionado según la estructura en árbol de tipos múltiples puede incluir dos tipos de división según una estructura en árbol binario y dos tipos de división según una estructura en árbol ternario.

Los dos tipos de división según la estructura en árbol binario pueden incluir división binaria vertical (SPLIT_BT_VER) y división binaria horizontal (SPLIT_BT_HOR). La división binaria vertical (SPLIT_BT_VER) significa que la CU actual se divide en dos partes iguales en la dirección vertical. Como se muestra en la FIG. 4, mediante división binaria vertical, se pueden generar dos CU que tengan la misma altura que la CU actual y que tengan una anchura que sea la mitad de la anchura de la CU actual. La división binaria horizontal (SPLIT_BT_HOR) significa que la CU actual se divide en dos partes iguales en dirección horizontal. Como se muestra en la FIG. 5, mediante división binaria horizontal, se pueden generar dos CU que tengan una altura que sea la mitad de la altura de la CU actual y que tengan la misma anchura que la CU actual.

Según la estructura en árbol ternario, dos tipos de división pueden incluir la división ternaria vertical (SPLIT_TT_VER) y la división ternaria horizontal (SPLIT_TT_HOR). En la división ternaria vertical (SPLIT_TT_VER), la CU actual se divide en dirección vertical en una proporción de 1:2:1. Como se muestra en la FIG. 5, mediante división ternaria vertical, se pueden generar dos CU que tengan la misma altura que la CU actual y que tengan una anchura que sea 1/4 de la anchura de la CU actual y una CU que tenga la misma altura que la CU actual y que tenga una anchura que sea la mitad de la anchura de la CU actual. En la división ternaria horizontal (SPLIT_TT_HOR), la CU actual se divide en dirección horizontal en una proporción de 1:2:1. Como se muestra en la FIG. 5, mediante división ternaria horizontal, se puede generar dos CU que tengan una altura que sea 1/4 de la altura de la CU actual y que tengan la misma anchura que la CU actual y una CU que tenga una altura que sea la mitad de la altura de la CU actual y que tenga la misma anchura que la CU actual.

La FIG. 6 es una vista que muestra un mecanismo de señalización de información de división de bloques en un árbol cuádruple con estructura en árbol de tipos múltiples anidado según la presente divulgación.

Aquí, la CTU se trata como el nodo raíz del árbol cuádruple y se subdivide por primera vez en una estructura en árbol cuádruple. Se señala la información (por ejemplo, `qt_split_flag`) que especifica si la división en árbol cuádruple se realiza en la CU actual (CTU o nodo (`QT_node`) del árbol cuádruple). Por ejemplo, cuando `qt_split_flag` tiene un primer valor (por ejemplo, "1"), la CU actual puede subdividirse en árbol cuádruple. Además, cuando `qt_split_flag` tiene un segundo valor (por ejemplo, "0"), la CU actual no se subdivide en árbol cuádruple, sino que se convierte en el nodo hoja (`QT_leaf_node`) del árbol cuádruple. Cada nodo hoja de árbol cuádruple puede luego subdividirse en estructuras en árbol de tipos múltiples. Es decir, el nodo hoja del árbol cuádruple puede convertirse en el nodo (`MTT_node`) del árbol de tipos múltiples. En la estructura en árbol de tipos múltiples, se señala una primera bandera (por ejemplo, `Mtt_split_cu_flag`) para especificar si el nodo actual se subdivide adicionalmente. Si el nodo correspondiente se subdivide adicionalmente (por ejemplo, si la primera bandera es 1), se puede señalar una segunda bandera (por ejemplo, `Mtt_split_cu_vertical_flag`) para especificar la dirección de división. Por ejemplo, la dirección de división puede ser una dirección vertical si la segunda bandera es 1 y puede ser una dirección horizontal si la segunda bandera es 0. Luego, se puede señalar una tercera bandera (por ejemplo, `Mtt_split_cu_binary_flag`) para especificar si el tipo de división es un tipo de división binaria o un tipo de división ternaria. Por ejemplo, el tipo de división puede ser un tipo de división binaria cuando la tercera bandera es 1 y puede ser un tipo de división ternaria cuando la tercera bandera es 0. El nodo del árbol de tipos múltiples adquirido mediante división binaria o división ternaria se puede subdividir adicionalmente en estructuras en árbol de tipos múltiples. Sin embargo, el nodo del árbol de tipos múltiples no puede subdividirse en estructuras de árbol cuádruple. Si la primera bandera es 0, el nodo correspondiente del árbol de tipos múltiples ya no se subdivide sino que se convierte en el nodo hoja (`MTT_leaf_node`) del árbol de tipos múltiples. La CU correspondiente al nodo hoja del árbol de tipos múltiples se puede utilizar como la unidad de codificación final descrita anteriormente.

Basándose en la `mtt_split_cu_vertical_flag` y la `mtt_split_cu_binary_flag`, se puede derivar un modo de división en árbol de tipos múltiples (`MttSplitMode`) de una CU como se muestra en la Tabla 1 a continuación. En la siguiente descripción, el modo de división en árbol de tipos múltiples puede denominarse tipo de división en árboles múltiples o tipo de división.

[Tabla 1]

MttSplitMode	mtt_split_cu_vertical_flag	mtt_split_cu_binary_flag
SPLIT_TT_HOR	0	0
SPLIT_BT_HOR	0	1
SPLIT_TT_VER	1	0
SPLIT_BT_VER	1	1

La FIG. 7 es una vista que muestra un ejemplo en el que una CTU se subdivide en múltiples CU aplicando un árbol de tipos múltiples después de aplicar un árbol cuádruple. En la Fig. 7, los bordes de bloque en negrita 710 representan particionado en árbol cuádruple y los bordes restantes 720 representan particionado en árbol de tipos múltiples. La CU puede corresponder a un bloque de codificación (CB). En una realización, la CU puede incluir un bloque de codificación de muestras de luminancia y dos bloques de codificación de muestras de crominancia correspondientes a las muestras de luminancia. Se puede derivar un tamaño de CB o TB de la componente de crominancia (muestra) basándose en un tamaño de CB o TB de la componente de luminancia (muestra) de acuerdo con la proporción de componentes según el formato de color (formato de crominancia, por ejemplo, 4:4:4, 4:2:2, 4:2:0 o similar) del fotograma/imagen. En el caso del formato de color 4:4:4, el tamaño de CB/TB de la componente de crominancia se puede establecer igual al tamaño de CB/TB de la componente de luminancia. En el caso del formato de color 4:2:2, la anchura del CB/TB de la componente de crominancia se puede establecer en la mitad de la anchura del CB/TB de la componente de luminancia y la altura del CB/TB de la componente de crominancia se puede establecer en la altura del CB/TB de la componente de luminancia. En el caso del formato de color 4:2:0, la anchura del CB/TB de la componente de crominancia se puede establecer en la mitad de la anchura del CB/TB de la componente de luminancia y la altura del CB/TB de la componente de crominancia se puede establecer en la mitad de la altura del CB/TB de la componente de luminancia.

En una realización, cuando el tamaño de la CTU es 128 basándose en la unidad de muestra de luminancia, el tamaño de la CU puede tener un tamaño de 128x128 a 4x4 que es el mismo tamaño que la CTU. En una realización, en el caso de un formato de color 4:2:0 (o formato de crominancia), un tamaño de CB de crominancia puede tener un tamaño de 64x64 a 2x2.

Mientras tanto, en una realización, el tamaño de CU y el tamaño de TU pueden ser iguales. Alternativamente, puede haber una pluralidad de TU en una región de CU. El tamaño de TU generalmente representa el tamaño de un bloque de transformada (TB) de componente de luminancia (muestra).

El tamaño de TU se puede derivar basándose en el tamaño de TB más grande permitido, `maxTbSize`, que es un valor predeterminado. Por ejemplo, cuando el tamaño de CU es mayor que `maxTbSize`, se puede derivar una pluralidad de TU (TB) que tienen `maxTbSize` a partir de la CU y la transformada/transformada inversa se puede realizar en unidades de TU (TB). Por ejemplo, el tamaño de TB de luminancia más grande permitido puede ser 64×64 y el tamaño de TB de crominancia más grande permitido puede ser 32×32 . Si la anchura o altura del CB subdividido según la estructura en árbol es mayor que la anchura o altura de transformada más grande, el CB puede subdividirse automáticamente (o implícitamente) hasta que se cumpla el límite de tamaño de TB en las direcciones horizontal y vertical.

Además, por ejemplo, cuando se aplica la predicción intra, se puede derivar un modo/tipo de predicción intra en unidades de CU (o CB) y un procedimiento de derivación de muestras de referencia vecinas y generación de muestras de predicción se puede realizar en unidades de TU (o TB). En este caso, puede haber una o una pluralidad de TU (o TB) en una región de CU (o CB) y, en este caso, la pluralidad de TU o (TB) puede compartir el mismo modo/tipo de predicción intra.

Mientras tanto, para un esquema de árbol de codificación de árbol cuádruple con árbol de tipos múltiples anidado, los siguientes parámetros pueden señalizarse como elementos de sintaxis de SPS desde el aparato de codificación al aparato de decodificación. Por ejemplo, se señala al menos uno de un tamaño de CTU, que es un parámetro que representa el tamaño del nodo raíz de un árbol cuádruple, `MinQTSize`, que es un parámetro que representa el tamaño mínimo permitido del nodo hoja del árbol cuádruple, `MaxBtSize`, que es un parámetro que representa el tamaño máximo permitido del nodo raíz del árbol binario, `MaxTtSize`, que es un parámetro que representa el tamaño máximo permitido del nodo raíz del árbol ternario, `MaxMttDepth`, que es un parámetro que representa la profundidad de jerarquía máxima permitida de la división en árbol de tipos múltiples desde un nodo hoja de árbol cuádruple, `MinBtSize`, que es un parámetro que representa el tamaño mínimo permitido del nodo hoja del árbol binario, o `MinTtSize`, que es un parámetro que representa el tamaño mínimo permitido del nodo hoja del árbol ternario.

Como una realización del uso del formato de crominancia 4:2:0, el tamaño de CTU se puede establecer en bloques de luminancia de 128×128 y dos bloques de crominancia de 64×64 correspondientes a los bloques de luminancia. En este caso, `MinOTSize` se puede establecer en 16×16 , `MaxBtSize` se puede establecer en 128×128 , `MaxTtSize` se puede establecer en 64×64 , `MinBtSize` y `MinTtSize` se pueden establecer en 4×4 y `MaxMttDepth` se puede establecer en 4. El particionado en árbol cuádruple se puede aplicar a la CTU para generar nodos hoja de árbol cuádruple. El nodo hoja del árbol cuádruple puede denominarse nodo hoja QT. Los nodos hoja de árbol cuádruple pueden tener un tamaño desde un tamaño de 16×16 (por ejemplo, el `MinOTSize`) hasta un tamaño de 128×128 (por ejemplo, el tamaño de CTU). Si el nodo hoja QT es de 128×128 , no se puede subdividir adicionalmente en un árbol binario/árbol ternario. Esto se debe a que, en este caso, incluso si se subdivide, excede `MaxBtSize` y `MaxTtSize` (por ejemplo, 64×64). En otros casos, los nodos hoja QT se pueden subdividir aún más en un árbol de tipos múltiples. Por lo tanto, el nodo hoja QT es el nodo raíz para el árbol de tipos múltiples, y el nodo hoja QT puede tener un valor 0 de profundidad de árbol de tipos múltiples (`mttDepth`). Si la profundidad del árbol de tipos múltiples alcanza `MaxMttDepth` (por ejemplo, 4), es posible que no se consideren particionados adicionales. Si la anchura del nodo del árbol de tipos múltiples es igual a `MinBtSize` y menor o igual que $2 \times \text{MinTtSize}$, entonces no se podrá considerar ningún particionado horizontal adicional. Si la altura del nodo del árbol de tipos múltiples es igual a `MinBtSize` y menor o igual que $2 \times \text{MinTtSize}$, no se podrá considerar ningún particionado vertical adicional. Cuando no se considera el particionado, el aparato de codificación puede saltar la señalización de información de particionado. En este caso, el aparato de decodificación puede derivar información de particionado con un valor predeterminado.

Mientras tanto, una CTU puede incluir un bloque de codificación de muestras de luminancia (denominado en lo sucesivo "bloque de luminancia") y dos bloques de codificación de muestras de crominancia correspondientes al mismo (denominado en lo sucesivo "bloques de crominancia"). El esquema de árbol de codificación descrito anteriormente puede aplicarse igualmente o por separado al bloque de luminancia y al bloque de crominancia de la CU actual. Específicamente, los bloques de luminancia y crominancia en una CTU se pueden subdividir en la misma estructura en árbol de bloques y, en este caso, la estructura en árbol se representa como `SINGLE_TREE`. Alternativamente, los bloques de luminancia y crominancia en una CTU pueden subdividirse en estructuras en árbol de bloques separadas y, en este caso, la estructura en árbol puede representarse como `DUAL_TREE`. Es decir, cuando la CTU se subdivide en árboles duales, la estructura en árbol de bloques para el bloque de luminancia y la estructura en árbol de bloques para el bloque de crominancia pueden estar presentes por separado. En este caso, la estructura en árbol de bloques para el bloque de luminancia se puede llamar `DUAL_TREE_LUMA`, y la estructura en árbol de bloques para la componente de crominancia se puede llamar `DUAL_TREE_CHROMA`. Para los grupos de porciones/mosaicos P y B, los bloques de luminancia y crominancia en una CTU pueden limitarse a tener la misma estructura en árbol de codificación. Sin embargo, para los grupos de porciones/mosaicos I, los bloques de luminancia y crominancia pueden tener estructuras en árbol de bloques separadas entre sí. Si se aplica la estructura en árbol de bloques separada, el CTB de luminancia puede subdividirse en CU basadas en una estructura de árbol de codificación particular, y el CTB de crominancia puede subdividirse en CU de crominancia basadas en otra estructura de árbol de codificación. Es decir, esto significa que una CU en un grupo de porciones/mosaicos I, al que se aplica la estructura en árbol de bloques separada, puede incluir un bloque de codificación de componentes de luminancia o bloques de codificación de dos componentes de crominancia y una CU de un grupo de porciones/mosaicos P o B puede incluir bloques de tres componentes de color (una componente de luminancia y dos componentes de crominancia).

Aunque se ha descrito una estructura de árbol de codificación de árbol cuádruple con un árbol de tipos múltiples

anidado, una estructura en la que se subdivide una CU no se limita a la misma. Por ejemplo, la estructura de BT y la estructura de TT pueden interpretarse como un concepto incluido en una estructura de árbol de particionado múltiple (MPT), y la CU puede interpretarse como subdividida a través de la estructura de QT y la estructura de MPT. En un ejemplo donde la CU se subdivide a través de una estructura de QT y una estructura de MPT, se puede señalar un elemento de sintaxis (por ejemplo, `MPT_split_type`) que incluye información sobre en cuántos bloques se subdivide el nodo hoja de la estructura de QT y un elemento de sintaxis (por ejemplo, `MPT_split_mode`) que incluye información sobre en cuál de las direcciones vertical y horizontal se subdivide el nodo hoja de la estructura de QT para determinar una estructura de particionado.

En otro ejemplo, la CU puede subdividirse de una manera diferente a la estructura de QT, estructura de BT o estructura de TT. Es decir, a diferencia de la CU de menor profundidad que se subdivide en 1/4 de la CU de mayor profundidad según la estructura de QT, la CU de menor profundidad se subdivide en 1/2 de la CU de mayor profundidad según la estructura de BT, o la CU de menor profundidad se subdivide en 1/4 o 1/2 de la CU de mayor profundidad según la estructura de TT, la CU de menor profundidad se puede subdividir en 1/5, 1/3, 3/8, 3/5, 2/3 o 5/8 de la CU de mayor profundidad en algunos casos, y el método de particionado de la CU no se limita a ello.

La estructura de bloques de codificación en árbol cuádruple con el árbol de tipos múltiples puede proporcionar una estructura de particionado de bloques muy flexible. Debido a los tipos de partición admitidos en un árbol de tipos múltiples, diferentes patrones de partición pueden dar como resultado la misma estructura de bloque de codificación en algunos casos. En el aparato de codificación y el aparato de decodificación, al limitar la aparición de dichos patrones de partición redundantes, se puede reducir la cantidad de datos de información de particionado.

Por ejemplo, la FIG. 8 muestra patrones de división redundantes que pueden ocurrir en la división en árbol binario y en la división en árbol ternario. Como se muestra en la FIG. 8, la división binaria continua 810 y 820 para una dirección de niveles de dos pasos tiene la misma estructura de bloques de codificación que la división binaria para una partición central después de la división ternaria. En este caso, puede prohibirse la división en árbol binario para los bloques centrales 830 y 840 de división en árbol ternario. Esta prohibición es aplicable a las CU de todos los fotogramas. Cuando dicha división específica está prohibida, la señalización de los elementos de sintaxis correspondientes podrá modificarse reflejando este caso prohibido, reduciendo así el número de bits señalizados para la división. Por ejemplo, como se muestra en el ejemplo mostrado en la FIG. 8, cuando está prohibida la división en árbol binario para el bloque central de la CU, no se señala un elemento de sintaxis `mtt_split_cu_binary_flag` que especifica si la división es una división binaria o una división ternaria y el valor del mismo puede derivarse como 0 mediante un aparato de decodificación.

Descripción general del formato de crominancia

A continuación se describirá un formato de crominancia. Una imagen puede codificarse en datos codificados que incluyen una matriz de componentes de luminancia (por ejemplo, Y) y dos matrices de componentes de crominancia (por ejemplo, Cb y Cr). Por ejemplo, un píxel de la imagen codificada puede incluir una muestra de luminancia y una muestra de crominancia. Se puede usar un formato de crominancia para representar un formato de configuración de la muestra de luminancia y la muestra de crominancia, y el formato de crominancia se puede denominar formato de color.

En una realización, una imagen puede codificarse en varios formatos de crominancia, tales como 4:2:0, 4:2:2 o 4:4:4 monocromo. En el muestreo monocromo, puede haber una matriz de muestras y la matriz de muestras puede ser una matriz de luminancia. En el muestreo 4:2:0, puede haber una matriz de muestras de luminancia y dos matrices de muestras de crominancia, cada una de las dos matrices de crominancia puede tener una altura igual a la mitad de aquella de la matriz de luminancia y una anchura igual a la mitad de aquella de la de la matriz de luminancia. En el muestreo 4:2:2, puede haber una matriz de muestras de luminancia y dos matrices de muestras de crominancia, cada una de las dos matrices de crominancia puede tener una altura igual a la de la matriz de luminancia y una anchura igual a la mitad de aquella de la matriz de luminancia. En el muestreo 4:4:4, puede haber una matriz de muestras de luminancia y dos matrices de muestras de crominancia, y cada una de las dos matrices de crominancia puede tener una altura y una anchura iguales a las de la matriz de luminancia.

Por ejemplo, en el muestreo 4:2:0, una muestra de crominancia puede ubicarse debajo de una muestra de luminancia correspondiente a la misma. En el muestreo 4:2:2, se puede ubicar una muestra de crominancia para superponerse a una muestra de luminancia correspondiente a la misma. En el muestreo 4:4:4, tanto una muestra de luminancia como una muestra de crominancia pueden ubicarse en una posición superpuesta.

Un formato de crominancia usado en un aparato de codificación y un aparato de decodificación puede estar predeterminado. Alternativamente, se puede señalar un formato de crominancia desde un aparato de codificación a un aparato de decodificación para ser utilizado de forma adaptativa en el aparato de codificación y el aparato de decodificación. En una realización, el formato de crominancia se puede señalar basándose en al menos uno de `chroma_format_idc` o `separate_colour_plane_flag`. Al menos uno de `chroma_format_idc` o `separate_colour_plane_flag` se puede señalar mediante una sintaxis de nivel superior, como DPS, VPS, SPS o PPS. Por ejemplo, `chroma_format_idc` y `separate_colour_plane_flag` pueden incluirse en la sintaxis SPS que se muestra en la FIG. 9.

Mientras tanto, la FIG. 10 muestra una realización de clasificación de formato de crominancia utilizando la señalización de `chroma_format_idc` y `separate_colour_plane_flag`. `chroma_format_idc` puede ser información que especifica un

formato de crominancia que se aplica a una imagen codificada. `separate_colour_plane_flag` puede especificar si una matriz de colores se procesa por separado en un formato de crominancia específico. Por ejemplo, un primer valor (por ejemplo, 0) de `chroma_format_idc` puede especificar muestreo monocromo. Un segundo valor (por ejemplo, 1) de `chroma_format_idc` puede especificar un muestreo 4:2:0. Un tercer valor (por ejemplo, 2) de `chroma_format_idc` puede especificar muestreo 4:2:2. Un cuarto valor (por ejemplo, 3) de `chroma_format_idc` puede especificar muestreo 4:4:4.

En 4:4:4, se puede aplicar lo siguiente según el valor de `separate_colour_plane_flag`. Si el valor de `separate_colour_plane_flag` es un primer valor (por ejemplo, 0), cada una de las dos matrices de crominancia puede tener la misma altura y anchura que una matriz de luminancia. En este caso, un valor de `ChromaArrayType` que especifica un tipo de matriz de muestras de crominancia se puede establecer igual a `chroma_format_idc`. Si el valor de `separate_colour_plane_flag` es un segundo valor (por ejemplo, 1), las matrices de muestras de luminancia, Cb y Cr pueden procesarse por separado y procesarse junto con fotogramas muestreados monocromos. En este caso, `ChromaArrayType` puede establecerse en 0.

Predicción intra en bloque de crominancia

Cuando se realiza una predicción intra en un bloque actual, se puede realizar una predicción en un bloque de componente de luminancia (bloque de luminancia) del bloque actual y una predicción en un bloque de componente de crominancia (bloque de crominancia). En este caso, el modo de predicción intra para el bloque de crominancia se puede establecer por separado del modo de predicción intra para el bloque de luminancia.

Por ejemplo, el modo de predicción intra para el bloque de crominancia se puede especificar basándose en la información de modo de predicción de crominancia intra, y la información de modo de predicción de crominancia intra se puede señalar en forma de un elemento de sintaxis `intra_chroma_pred_mode`. Por ejemplo, la información de modo de predicción de crominancia intra puede representar uno de un modo plano, un modo DC, un modo vertical, un modo horizontal, un modo derivado (DM) y un modo de modelo lineal de componentes cruzadas (CCLM). Aquí, el modo plano puede especificar el modo de predicción intra #0, el modo DC puede especificar el modo de predicción intra #1, el modo vertical puede especificar el modo de predicción intra #26 y el modo horizontal puede especificar el modo de predicción intra #10. DM también puede denominarse modo directo. El CCLM también puede denominarse modelo lineal (LM). El modo CCLM puede incluir al menos uno de `L_CCLM`, `T_CCLM` y `LT_CCLM`.

Mientras tanto, el DM y el CCLM son modos de predicción intra dependientes para predecir el bloque de crominancia utilizando información sobre el bloque de luminancia. El DM puede representar un modo en el que el mismo modo de predicción intra que el modo de predicción intra para la componente de luminancia se aplica como modo de predicción intra para la componente de crominancia. Además, el CCLM puede representar un modo de predicción intra usando, como muestras de predicción del bloque de crominancia, muestras derivadas submuestreando muestras reconstruidas del bloque de luminancia y luego aplicando α y β que son parámetros de CCLM a muestras submuestreadas en un proceso de generación del bloque de predicción para el bloque de crominancia.

Modo CCLM (modelo lineal de componentes cruzadas)

Como se describió anteriormente, un modo CCLM puede aplicarse a un bloque de crominancia. El modo CCLM es un modo de predicción intra que utiliza correlación entre un bloque de luminancia y un bloque de crominancia correspondiente al bloque de luminancia, y se realiza derivando un modelo lineal basado en muestras vecinas del bloque de luminancia y muestras vecinas del bloque de crominancia. Además, se puede derivar una muestra de predicción del bloque de crominancia basándose en el modelo lineal derivado y las muestras reconstruidas del bloque de luminancia.

Específicamente, cuando el modo CCLM se aplica a un bloque de crominancia actual, los parámetros para un modelo lineal pueden derivarse basándose en muestras vecinas utilizadas para la predicción intra del bloque de crominancia actual y muestras vecinas utilizadas para la predicción intra de un bloque de luminancia actual. Por ejemplo, el modelo lineal para CCLM se puede expresar basándose en la siguiente ecuación.

[Ecuación 1]

$$pred_c(i,j) = \alpha \cdot rec_l(i,j) + \beta$$

donde, $pred_c(i,j)$ puede denotar la muestra de predicción de las coordenadas (i,j) del bloque de crominancia actual en la CU actual. $rec_l(i,j)$ puede denotar la muestra reconstruida de las coordenadas (i,j) del bloque de luminancia actual en la CU. Por ejemplo, $rec_l(i,j)$ puede indicar la muestra reconstruida con submuestreo del bloque de luminancia actual. Los coeficientes del modelo lineal α y β pueden señalizarse o derivarse de muestras vecinas.

Descripción general del modo de paleta

A continuación se describirá un modo de paleta. Un aparato de codificación según una realización puede codificar una imagen usando un modo de paleta, y un aparato de decodificación puede decodificar una imagen usando un modo de paleta de una manera correspondiente al mismo. El modo de paleta puede denominarse modo de codificación de paleta, modo de paleta intra, modo de codificación de paleta intra, etc. El modo de paleta puede considerarse como un tipo de modo de codificación intra o puede considerarse como uno de los métodos de predicción intra. Sin embargo,

al igual que en el modo de salto descrito anteriormente, es posible que no se señalice un valor residual separado para el bloque correspondiente.

En una realización, el modo de paleta se puede usar para mejorar la eficiencia de codificación al codificar contenido de pantalla que es una imagen generada por una computadora que incluye una cantidad significativa de texto y gráficos. En general, un área local de la imagen generada como contenido de la pantalla está separada por bordes nítidos y se expresa con un pequeño número de colores. Para utilizar esta característica, en el modo de paleta, las muestras de un bloque pueden expresarse mediante índices que especifican una entrada de color de la tabla de paleta.

Para aplicar un modo de paleta, se puede señalar información sobre una tabla de paleta. En una realización, la tabla de paleta puede incluir un valor de índice correspondiente a cada color. Para señalar el valor del índice, se puede señalar información de predicción del índice de paleta. La información de predicción del índice de paleta puede incluir un valor de índice para al menos una parte de un mapa de índices de paleta. En el mapa de índices de paleta, los píxeles de datos de vídeo se pueden mapear con índices de color de la tabla de paleta.

La información de predicción del índice de paleta puede incluir información del valor de ejecución. Para al menos una parte del mapa de índices de paleta, la información del valor de ejecución puede asociar un valor de ejecución con un valor de índice. Un valor de ejecución puede estar asociado con un índice de color de escape. El mapa de índices de paleta puede generarse a partir de la información de predicción del índice de paleta. Por ejemplo, al menos una parte del mapa de índices de paleta se puede generar determinando si se ajusta el valor de índice de la información de predicción del índice de paleta basándose en un último valor de índice.

Un bloque actual en un fotograma actual puede codificarse o reconstruirse según el mapa de índices de paleta. Al aplicar el modo de paleta, un valor de píxel en una unidad de codificación actual puede expresarse como un pequeño conjunto de valores de color representativos. A este conjunto se le puede llamar paleta. Para píxeles que tienen un valor cercano a un color de paleta, se puede señalar el índice de paleta. Para píxeles que tienen un valor que no pertenece a (está fuera de) la paleta, los píxeles correspondientes pueden denotarse mediante un símbolo de escape y se puede señalar directamente un valor de píxel cuantificado. En este documento, un píxel o un valor de píxel se pueden describir como una muestra.

Para codificar un bloque codificado en el modo de paleta, un aparato de decodificación puede decodificar un color de paleta y un índice. El color de paleta puede describirse en la tabla de paleta y puede codificarse utilizando una herramienta de codificación de tabla de paleta. Puede señalarse una bandera de escape para cada unidad de codificación. La bandera de escape puede especificar si un símbolo de escape está presente en una unidad de codificación actual. Si el símbolo de escape está presente, la tabla de paleta se puede aumentar en 1 unidad (por ejemplo, unidad de índice) y se puede designar un último índice como un modo de escape. Los índices de paleta de todos los píxeles para una unidad de codificación pueden configurar el mapa de índices de paleta y pueden codificarse usando una herramienta de codificación de mapa de índices de paleta.

Por ejemplo, para codificar la tabla de paleta, se puede mantener un predictor de paleta. El predictor de paleta se puede inicializar en un punto de inicio de cada porción. Por ejemplo, el predictor de paleta se puede restablecer a 0. Para cada entrada del predictor de paleta, se puede señalar una bandera de reutilización que especifica si es una parte de una paleta actual. La bandera de reutilización puede señalarse utilizando una codificación de longitud de ejecución de un valor de 0.

A partir de entonces, los números de las nuevas entradas de la paleta se pueden señalar utilizando un código Golomb exponencial de orden cero. Finalmente, se pueden señalar los valores de las componentes para una nueva entrada de paleta. Después de codificar una unidad de codificación actual, el predictor de paleta se puede actualizar usando la paleta actual, y una entrada de un predictor de paleta anterior que no se reutiliza en la paleta actual (hasta alcanzar un tamaño máximo permitido) se puede agregar al final de un nuevo predictor de paleta y esto puede denominarse relleno de paleta.

Por ejemplo, para codificar el mapa de índices de paleta, los índices pueden codificarse usando exploración horizontal o vertical. Se puede señalar un orden de exploración a través de un tren de bits utilizando `palette_transpose_flag`, que es un parámetro que especifica una dirección de exploración. Por ejemplo, cuando se aplica la exploración horizontal a índices de exploración para muestras en una unidad de codificación actual, `palette_transpose_flag` puede tener un primer valor (por ejemplo, 0) y cuando se aplica la exploración vertical, `palette_transpose_flag` puede tener un segundo valor (por ejemplo, 1). La FIG. 11 muestra una realización de exploración horizontal y exploración vertical según una realización.

Además, en una realización, el índice de paleta se puede codificar usando un modo 'INDEX' y un modo 'COPY_ABOVE'. Excepto para el caso en el que se señala el modo del índice de paleta para la fila superior cuando se utiliza la exploración horizontal, el caso en el que se señala el modo del índice de paleta para una columna situada más a la izquierda cuando se usa la exploración vertical y el caso en el que un modo inmediatamente anterior es 'COPY_ABOVE', los dos modos pueden señalarse usando una bandera.

En un modo 'INDEX', el índice de paleta puede señalarse explícitamente. Para un modo 'INDEX' y un modo 'COPY_ABOVE', se puede señalar un valor de ejecución que especifica el número de píxeles codificados usando el

mismo modo.

Se puede establecer un orden de codificación para un mapa de índices de la siguiente manera. En primer lugar, se puede señalar el número de valores de índice para una unidad de codificación. Esto se puede realizar después de la señalización de un valor de índice real para toda la unidad de codificación usando codificación binaria truncada. Tanto el número de índices como los valores de los índices pueden codificarse en modo de derivación (bypass). A través de esto, se pueden agrupar bins de derivación relacionados con el índice. Luego, el modo de paleta (INDEX o COPY_ABOVE) y el valor de ejecución se pueden señalar utilizando un método de entrelazado.

5 Finalmente, los valores de escape de componentes correspondientes a muestras de escape para toda la unidad de codificación pueden agruparse y codificarse mutuamente en un modo de derivación. last_run_type_flag, que es un elemento de sintaxis adicional, puede señalizarse después de señalar el valor del índice. Al utilizar last_run_type_flag junto con el número de índices, se puede saltar la señalización de un valor de ejecución correspondiente a una última ejecución en el bloque.

10 En una realización, se puede utilizar para una porción I un tipo de árbol dual, en el que se realiza un particionado de unidad de codificación independiente en una componente de luminancia y una componente de crominancia. El modo de paleta puede aplicarse a la componente de luminancia y a la componente de crominancia individual o conjuntamente. Si no se aplica el árbol dual, el modo de paleta se aplica a todas las componentes Y, Cb y Cr.

En una realización, un elemento de sintaxis para el modo de paleta puede codificarse y señalizarse como se muestra en las FIGS. 12 a 19. De la FIG. 12 a la FIG. 13 muestran sintaxis consecutivas en una unidad de codificación (CU) para un modo de paleta, y las FIGS. 14 a 19 muestran sintaxis consecutivas para un modo de paleta.

20 A continuación se describirá cada elemento de sintaxis. pred_mode_plt_flag, que es una bandera de modo de paleta, puede especificar si el modo de paleta se aplica a una unidad de codificación actual. Por ejemplo, un primer valor (por ejemplo, 0) de pred_mode_plt_flag puede especificar que el modo de paleta no se aplica a la unidad de codificación actual. Un segundo valor (por ejemplo, 1) de pred_mode_plt_flag puede especificar que el modo de paleta se aplica a la unidad de codificación actual. Cuando no se obtiene pred_mode_plt_flag de un tren de bits, se puede determinar que el valor de pred_mode_plt_flag es el primer valor.

Un parámetro PredictorPaletteSize[startComp] puede especificar el tamaño de una paleta predictora para startComp, que es una primera componente de color de una tabla de paleta actual.

30 Un parámetro PalettePredictorEntryReuseFlags[i] puede ser información que especifique si reutilizar una entrada. Por ejemplo, un primer valor (por ejemplo, 0) de PalettePredictorEntryReuseFlags[i] puede especificar que una i-ésima entrada de la paleta predictora puede especificar que la i-ésima entrada de la paleta predictora no es una entrada de una paleta actual y un segundo valor (por ejemplo, 1) puede especificar que la i-ésima entrada de la paleta predictora se puede reutilizar en la paleta actual. Para el uso de PalettePredictorEntryReuseFlags[i], se puede establecer un valor inicial en 0.

35 Un parámetro palette_predictor_run puede especificar el número de 0 presentes antes de una entrada distinta de cero en una matriz PalettePredictorEntryReuseFlags.

Un parámetro num_signalled_palette_entries puede especificar el número de entradas en la paleta actual señalizadas explícitamente para una primera componente de color startComp de la tabla de paleta actual. Cuando no se obtiene num_signalled_palette_entries del tren de bits, se puede determinar que el valor de num_signalled_palette_entries es 0.

40 Un parámetro CurrentPaletteSize[startComp] puede especificar el tamaño de la paleta actual para la primera componente de color startComp de la tabla de paleta actual. Esto se puede calcular como se muestra en la siguiente ecuación. El valor de CurrentPaletteSize[startComp] puede tener un rango de 0 a palette_max_size.

[Ecuación 2]

$$\text{CurrentPaletteSize[startComp]} = \text{NumPredictedPaletteEntries} + \text{num_signalled_palette_entries}$$

Un parámetro new_palette_entries[cldx][i] puede especificar un valor de una entrada de paleta señalizada i-ésima para una componente de color cldx.

45 Un parámetro PredictorPaletteEntries[cldx][i] puede especificar un i-ésimo elemento en una paleta predictora para la componente de color cldx.

Un parámetro CurrentPaletteEntries[cldx][i] puede especificar un i-ésimo elemento en la paleta actual para la componente de color cldx. PredictorPaletteEntries y CurrentPaletteEntries se pueden generar como se muestra en la ecuación de la FIG. 20.

50 Un parámetro palette_escape_val_present_flag puede especificar si está presente una muestra de codificación de

- 5 escape. Por ejemplo, un primer valor (por ejemplo, 0) de `palette_escape_val_present_flag` puede especificar que la muestra de codificación de escape está presente para la unidad de codificación actual, y un segundo valor (por ejemplo, 1) de `palette_escape_val_present_flag` puede especificar que la unidad de codificación actual incluye al menos una muestra de codificación de escape. Cuando no se obtiene `palette_escape_val_present_flag` del tren de bits, se puede determinar que el valor de `palette_escape_val_present_flag` es 1.
- Un parámetro `MaxPaletteIndex` puede especificar un valor máximo disponible de un índice de paleta para la unidad de codificación actual. Se puede determinar que el valor de `MaxPaletteIndex` es `CurrentPaletteSize[startComp] + palette_escape_val_present_flag`.
- 10 Un parámetro `num_palette_indices_minus1` puede especificar el número de índices de paleta señalizados explícita o implícitamente para un bloque actual. Por ejemplo, un valor obtenido sumando 1 a `num_palette_indices_minus1` puede especificar el número de índices de paleta señalizados explícita o implícitamente para el bloque actual. Cuando `num_palette_indices_minus1` no está incluido en el tren de bits, se puede determinar que el valor de `num_palette_indices_minus1` es 0.
- 15 Un parámetro `palette_idx_idc` puede ser un indicador de un índice para la tabla de paleta `CurrentPaletteEntries`. El valor de `palette_idx_idc` puede tener un valor de 0 a `MaxPaletteIndex` para un primer índice del bloque correspondiente, y puede tener un valor de 0 para `MaxPaletteIndex-1` para los índices restantes del bloque correspondiente. Cuando el valor de `palette_idx_idc` no se obtiene del tren de bits, se puede determinar que el valor de `palette_idx_idc` es 0.
- Un parámetro `PaletteIndexIdc[i]` puede ser una matriz que almacena el valor del i-ésimo `palette_idx_idc` señalado explícita o implícitamente. Los valores de todos los elementos de `PaletteIndexIdc[i]` se pueden inicializar a 0.
- 20 Un parámetro `copy_above_indices_for_final_run_flag` puede especificar información que especifica si se copia un índice anterior para una última ejecución, un primer valor (por ejemplo, 0) puede especificar que el índice de paleta en una última posición de la unidad de codificación actual se señala explícita o implícitamente a través del tren de bits, y un segundo valor (por ejemplo, 1) puede especificar que el índice de paleta en la última posición de la unidad de codificación actual se señala explícita o implícitamente a través del tren de bits. Cuando no se obtiene
- 25 `copy_above_indices_for_final_run_flag` del tren de bits, se puede determinar que el valor de `copy_above_indices_for_final_run_flag` es 0.
- Un parámetro `palette_transpose_flag` puede ser información que especifica un método de exploración usado para explorar un índice en busca de un píxel de una unidad de codificación actual. Por ejemplo, un primer valor (por ejemplo, 0) de `palette_transpose_flag` puede especificar que se aplica la exploración horizontal para explorar un índice en busca de un píxel de una unidad de codificación actual, y un segundo valor (por ejemplo, 1) de `palette_transpose_flag` puede especificar que se aplica la exploración vertical para explorar un índice en busca de un píxel de una unidad de codificación actual. Cuando no se obtiene `palette_transpose_flag` del tren de bits, se puede determinar que el valor de
- 30 `palette_transpose_flag` es 0.
- Un primer valor (por ejemplo, 0) de un parámetro `copy_above_palette_indices_flag` puede especificar que un indicador que especifica el índice de paleta de una muestra se obtiene o deriva de un valor codificado del tren de bits. Un segundo valor (por ejemplo, 1) de `copy_above_palette_indices_flag` puede especificar que el índice de paleta es el mismo que el índice de paleta de una muestra vecina. Por ejemplo, la muestra vecina puede ser una muestra ubicada en la misma posición que la muestra actual en una columna izquierda de la muestra actual cuando actualmente se está usando la exploración vertical. Alternativamente, la muestra vecina puede ser una muestra ubicada en la misma
- 35 posición que la muestra actual en una fila superior de la muestra actual cuando actualmente se está usando la exploración horizontal.
- 40 Un primer valor (por ejemplo, 0) de un parámetro `CopyAboveIndicesFlag[xC][yC]` puede especificar que el índice de paleta se obtiene explícita o implícitamente del tren de bits. Un segundo valor (por ejemplo, 1) puede especificar que se genera un índice de paleta copiando un índice de paleta de una columna izquierda cuando actualmente se está usando la exploración vertical o copiando un índice de paleta de una fila superior cuando actualmente se está usando la exploración horizontal. Aquí, `xC` e `yC` son indicadores de coordenadas que especifican relativamente la posición de la muestra actual desde la muestra superior izquierda del fotograma actual. `PaletteIndexMap[xC][yC]` puede tener un valor de 0 a $(\text{MaxPaletteIndex} - 1)$.
- 45 Un parámetro `PaletteIndexMap[xC][yC]` puede especificar un índice de paleta y, por ejemplo, especificar un índice para una matriz representada por `CurrentPaletteEntries`. Los índices de matriz `xC` e `yC` son indicadores de coordenadas que especifican las coordenadas de una muestra actual en relación con la muestra superior izquierda de un fotograma actual como se describió anteriormente. `PaletteIndexMap[xC][yC]` puede tener un valor de 0 a $(\text{MaxPaletteIndex} - 1)$.
- 50 Un parámetro `PaletteRun` puede especificar el número de posiciones consecutivas que tienen el mismo índice de paleta, cuando el valor de `CopyAboveIndicesFlag[xC][yC]` es 0. Mientras tanto, cuando el valor de `CopyAboveIndicesFlag[xC][yC]` es 1, `PaletteRun` puede especificar el número de posiciones consecutivas que tienen el mismo índice de paleta que un índice de paleta en una posición en una fila superior cuando la dirección de exploración actual es exploración horizontal y un índice de paleta en una posición en una columna izquierda cuando
- 55

la dirección de exploración actual es exploración vertical.

Un parámetro PaletteMaxRun puede especificar un valor máximo disponible de PaletteRun. El valor de PaletteMaxRun puede ser un número entero mayor que 0.

Un parámetro palette_run_prefix puede especificar una parte de prefijo utilizada para la binarización de PaletteRun.

- 5 Un parámetro palette_run_suffix puede especificar una parte de sufijo utilizada para la binarización de PaletteRun. Cuando no se obtiene palette_run_suffix de un tren de bits, se puede determinar que su valor es 0.

El valor de PaletteRun se puede determinar de la siguiente manera. Por ejemplo, cuando el valor de palette_run_prefix es menor que 2, este se puede calcular de la siguiente manera.

[Ecuación 3]

$$\text{PaletteRun} = \text{palette_run_prefix}$$

- 10 En caso contrario, cuando el valor de palette_run_prefix es igual o mayor que 2, este se puede calcular de la siguiente manera.

[Ecuación 4]

$$\text{PrefixOffset} = 1 \ll (\text{palette_run_prefix} - 1)$$

$$\text{PaletteRun} = \text{PrefixOffset} + \text{palette_run_suffix}$$

- 15 Un parámetro palette_escape_val puede especificar un valor de muestra de codificación de escape cuantificado para una componente. Un parámetro PaletteEscapeVal[cldx][xC][yC] puede especificar un valor de escape de una muestra con un valor de PaletteIndexMap[xC][yC] de (MaxPaletteIndex - 1) y un valor de palette_escape_val_present_flag de 1. Aquí, cldx puede especificar una componente de color. Los indicadores de matriz xC e yC pueden ser indicadores de posición que especifican distancias relativas de la posición de la muestra actual desde la muestra superior izquierda del fotograma actual, como se describió anteriormente.

Señalización del modo de predicción de crominancia al aplicar el modo de paleta

- 20 De aquí en adelante, se describirá un método para señalar información del modo de predicción de crominancia cuando se aplica un modo de paleta. En una realización, puede no aplicarse codificación de predicción de crominancia tal como CCLM a una unidad de codificación (o bloque de codificación) a la que se aplica el modo de paleta. Además, es posible que intra_chroma_pred_mode no se señalice para una unidad de codificación a la que se aplica el modo de paleta.

- 25 Al determinar que CCLM está disponible para una componente de crominancia, se puede señalar una bandera de activación/desactivación para el mismo. En una realización, la disponibilidad de CCLM se puede determinar usando sps_palette_enabled_flag o sps_plt_enabled_flag, y la bandera de activación/desactivación de CCLM se puede señalar usando sps_palette_enabled_flag o sps_plt_enabled_flag.

- 30 Mientras tanto, en el ejemplo de la FIG. 13, la señalización de información de CCLM (por ejemplo, cclm_mode_flag) no considera si se aplica un modo de paleta de una unidad de codificación. Por ejemplo, en el ejemplo de la FIG. 13, se describe una realización en la que se señala información de predicción de crominancia predeterminada (por ejemplo, cclm_mode_flag, intra_chroma_pred_mode) cuando un modo de paleta no se aplica a una unidad de codificación actual o cuando una unidad de codificación no es una crominancia de árbol dual.

- 35 En este caso, como la información de predicción de crominancia predeterminada se señala en un bloque codificado por paleta en un único árbol, puede producirse una señalización de sintaxis innecesaria. Además, a medida que se señala la información de predicción de crominancia predeterminada, incluso aunque una componente de crominancia en una unidad de codificación esté codificada en un modo de paleta, se puede realizar la predicción intra de crominancia según CCLM y se puede realizar la predicción intra de crominancia según un modo DM.

- 40 Para resolver el problema descrito anteriormente, se puede modificar la sintaxis para una unidad de codificación como se muestra en la FIG. 21. La FIG. 21 muestra la sintaxis de una unidad de codificación que especifica que la información de predicción intra de crominancia predeterminada 2120 se obtiene de un tren de bits, cuando un valor de pred_mode_plt_flag 2110, que es un parámetro que especifica si se aplica un modo de paleta a una unidad de codificación actual, es un primer valor (por ejemplo, 0) que especifica que el modo de paleta no se aplica. Además, la sintaxis de la FIG. 21 especifica que la información de predicción intra de crominancia predeterminada 2120 no se obtiene del tren de bits cuando el valor de pred_mode_plt_flag 2110 es un segundo valor (por ejemplo, 1) que especifica que se aplica el modo de paleta.

- 45 Como en la realización de la FIG. 21, para señalar la información de predicción de crominancia predeterminada (por ejemplo, cclm_mode_flag, intra_chroma_pred_mode), la información de predicción de crominancia predeterminada (por ejemplo, cclm_mode_flag, intra_chroma_pred_mode) se puede señalar dependiendo de si el modo de paleta se aplica a la unidad de codificación actual.

En lo sucesivo, la señalización de información de predicción intra de crominancia predeterminada según la sintaxis de la FIG. 21 se describirá con referencia a la FIG. 22. Un aparato de codificación o aparato de decodificación según una realización puede determinar si un modo de paleta se aplica a una unidad de codificación actual (por ejemplo, un bloque de codificación) (S2210). Por ejemplo, el aparato de decodificación puede determinar si el modo de paleta se aplica a la unidad de codificación actual según un valor de `pred_mode_plt_flag`.

A continuación, cuando el modo de paleta se aplica a la unidad de codificación actual, el aparato de codificación puede codificar la unidad de codificación correspondiente en el modo de paleta, y el aparato de decodificación puede decodificar la unidad de codificación correspondiente en el modo de paleta. Por lo tanto, el aparato de codificación o el aparato de decodificación puede no señalar información de predicción de crominancia predeterminada (S2220). Por ejemplo, el aparato de codificación puede no codificar la información de predicción de crominancia predeterminada (por ejemplo, `cclm_mode_flag`, `intra_chroma_pred_mode`), y el aparato de decodificación puede no obtener la información de predicción de crominancia predeterminada de un tren de bits.

A continuación, cuando el modo de paleta no se aplica a la unidad de codificación actual, el aparato de codificación o el aparato de decodificación puede señalar la información de predicción de crominancia predeterminada. En una realización, el aparato de codificación o el aparato de decodificación puede determinar si un modo CCLM está disponible para la unidad de codificación actual (S2230), señalar un parámetro de CCLM si está disponible (S2240) y señalar un parámetro `intra_chroma_pred_mode` si no está disponible (S2250).

Para ello, se describirá con mayor detalle el paso de obtener información de predicción de crominancia mediante el aparato de decodificación con referencia a la FIG. 23. Cuando el modo de paleta no se aplica a la unidad de codificación actual (S2310), el aparato de decodificación puede determinar si el modo CCLM está disponible para la unidad de codificación actual (S2320). Por ejemplo, un parámetro `sps_cclm_enabled_flag` que especifica la disponibilidad del modo CCLM señalado en un conjunto de parámetros de secuencia tiene un primer valor (por ejemplo, 0) que especifica que el modo CCLM no está disponible, el aparato de decodificación puede determinar que el modo CCLM no está disponible para la unidad de codificación actual. Alternativamente, cuando `sps_cclm_enabled_flag` tiene un segundo valor (por ejemplo, 1) que especifica que el modo `cclm` está disponible, un parámetro de tipo de porción `sh_slice_type` transmitido a través de una cabecera de porción especifica que un tipo de porción actual no es una porción I o un tamaño de una componente de luminancia de un bloque actual es menor que 64, el aparato de decodificación puede determinar que el modo CCLM está disponible para la unidad de codificación actual. Alternativamente, cuando `sps_cclm_enabled_flag` tiene un segundo valor (por ejemplo, 1) que especifica que el modo `cclm` está disponible, cada unidad de árbol de codificación (CTU) en la que un parámetro `sps_qtbt_dual_tree_intra_flag` señalado en un conjunto de parámetros de secuencia se incluye en una porción I se divide en bloques de componente de luminancia que tienen un tamaño de 64×64 , y no indica que cada CTU se convierta en un nodo de cabecera de un árbol dual, el aparato de decodificación puede determinar que el modo CCLM está disponible para la unidad de codificación actual.

Cuando el modo CCLM está disponible, el aparato de decodificación puede determinar si el CCLM se aplica a la unidad de codificación actual (S2330). Por ejemplo, el aparato de decodificación puede obtener un parámetro `cclm_mode_flag` del tren de bits. El parámetro `cclm_mode_flag` puede especificar si se aplica el modo CCLM. Un primer valor (por ejemplo, 0) de `cclm_mode_flag` puede especificar que el modo CCLM no se aplica. Un segundo valor (por ejemplo, 1) de `cclm_mode_flag` puede especificar que el modo CCLM de cualquiera de `T_CCLM`, `L_CCLM` y `LT_CCLM` es aplicable. Cuando el valor de `cclm_mode_flag` no se obtiene del tren de bits, se puede determinar que el valor de `cclm_mode_flag` es 0.

Cuando se aplica el modo CCLM (por ejemplo, `cclm_mode_flag == 1`), el aparato de decodificación puede obtener un parámetro `cclm_mode_idx` del tren de bits (S2340). El parámetro `cclm_mode_idx` puede especificar un índice que especifica un modo CCLM utilizado para decodificar una componente de crominancia de una unidad de codificación actual entre `T_CCLM`, `L_CCLM` y `LT_CCLM`.

Mientras tanto, cuando el modo CCLM no está disponible o el modo CCLM no se aplica (por ejemplo, `cclm_mode_flag == 0`), el aparato de decodificación puede obtener un parámetro `intra_chroma_pred_mode` del tren de bits (S2350). Como se describió anteriormente, el parámetro `intra_chroma_pred_mode` puede especificar un modo de predicción intra usado para decodificar una componente de crominancia de una unidad de codificación actual. Por ejemplo, `intra_chroma_pred_mode` puede especificar uno de un modo plano, un modo DC, un modo vertical, un modo horizontal y un modo derivado (DM).

Método de codificación

En lo sucesivo, se describirá con referencia a la FIG. 24 un método para realizar la codificación mediante un aparato de codificación según una realización utilizando el método descrito anteriormente. El aparato de codificación según una realización incluye una memoria y al menos un procesador, y el al menos un procesador puede realizar el siguiente método de codificación.

Primero, el aparato de codificación puede determinar un bloque actual dividiendo una imagen (S2410). Por ejemplo, el aparato de codificación puede determinar el bloque actual dividiendo la imagen como se describió anteriormente

con referencia a las FIGS. 4 a 6. En un proceso de división según una realización, la información de división de imagen puede codificarse y la información de división de imagen codificada puede generarse como un tren de bits.

5 A continuación, el aparato de codificación puede determinar un modo de predicción del bloque actual (S2420). A continuación, el aparato de codificación puede codificar una bandera de modo de paleta (por ejemplo, `pred_mode_plt_flag`) que especifica que el modo de predicción del bloque actual es un modo de paleta, en función de si el modo de predicción del modo actual es un modo de paleta (S2430). La bandera de modo de paleta codificada puede generarse como un tren de bits.

10 A continuación, el aparato de codificación puede codificar información de codificación de modo de paleta en la que el bloque actual se codifica en el modo de paleta, basándose en un tipo de árbol del bloque actual y en si el modo de predicción del bloque actual es un modo de paleta (S2440). Por ejemplo, al determinar que se aplica el modo de paleta, el aparato de codificación puede generar un tren de bits generando información de codificación de modo de paleta usando la sintaxis de `palette_coding()` como se describe con referencia a las FIGS. 14 a 19.

15 Mientras tanto, codificar la información de codificación de modo de paleta del bloque actual puede incluir codificar la información de codificación de modo de paleta de una componente de luminancia del bloque actual. Por ejemplo, cuando el tipo de árbol del bloque actual es un tipo de árbol único o un tipo de luminancia de árbol dual y el modo de paleta se aplica para el bloque actual, se puede codificar información para la predicción del modo de paleta de la componente de luminancia del bloque actual y se puede generar un tren de bits usando la información codificada.

20 En este caso, la información de codificación de modo de paleta de la componente de luminancia del bloque actual puede codificarse basándose en un tamaño del bloque de la componente de luminancia del bloque actual. Por ejemplo, como en la realización de la FIG. 13, el aparato de codificación puede generar la información de codificación de modo de paleta como un tren de bits de acuerdo con la sintaxis `palette_coding()` definida basándose en una anchura (por ejemplo, `cbWidth`) del bloque de la componente de luminancia del bloque actual y una altura (por ejemplo, `cbHeight`) del bloque de la componente de luminancia del bloque actual.

25 Además, codificar la información de codificación de modo de paleta del bloque actual puede incluir además codificar la información de codificación de modo de paleta de una componente de crominancia del bloque actual. Por ejemplo, cuando se aplica el modo de paleta para el bloque actual y un tipo de árbol del bloque actual es un tipo de crominancia de árbol dual, se puede codificar información para la predicción de modo de paleta de la componente de crominancia del bloque actual y se puede generar un tren de bits utilizando la información codificada.

30 En este caso, la información de codificación de modo de paleta de la componente de crominancia del bloque actual puede codificarse basándose en un tamaño del bloque de la componente de crominancia del bloque actual. Por ejemplo, como en la realización de la FIG. 13, el aparato de codificación puede generar la información de codificación de modo de paleta como un tren de bits, de acuerdo con la sintaxis de `palette_coding()` definida basándose en una anchura (por ejemplo, `cbWidth/subWidthC`) del bloque de la componente de crominancia del bloque actual y una altura (por ejemplo, `cbHeight/subHeightC`) del bloque de la componente de crominancia del bloque actual. Aquí, `subWidthC` y `subHeightC` pueden ser proporciones de la altura y la anchura del bloque de la componente de luminancia con respecto al bloque de la componente de crominancia. En una realización, `subWidthC` y `subHeightC` se pueden determinar basándose en `chroma_format_idc` y `separate_colour_plane_flag` como se muestra en la FIG. 10.

40 A continuación, cuando el modo de predicción del bloque actual no es un modo de paleta, el aparato de codificación puede codificar información de predicción de la componente de crominancia del bloque actual (S2450). La información de predicción de la componente de crominancia puede ser información (por ejemplo, `cclm_mode_flag`, `cclm_mode_idx`) para la predicción CCLM (modelo lineal de componentes cruzadas) o información de predicción intra de la componente de crominancia (por ejemplo, `intra_chroma_pred_mode`).

Mientras tanto, cuando el modo de paleta se aplica al bloque actual, es posible que el aparato de codificación no codifique la información de predicción de la componente de crominancia.

45 Más específicamente, la información sobre la predicción CCLM puede incluir una bandera de CCLM (por ejemplo, `cclm_mode_flag`) que especifica si se realiza predicción CCLM y un índice de modo CCLM (por ejemplo, `cclm_mode_idx`) que especifica un modo de predicción CCLM. La bandera de CCLM puede codificarse y generarse como un tren de bits, cuando la predicción CCLM está disponible para el bloque actual. El índice de modo CCLM puede codificarse y generarse como el tren de bits, cuando la bandera de CCLM especifica que se realiza la predicción CCLM.

50 Mientras tanto, cuando la bandera de CCLM especifica que no se realiza predicción CCLM, la información de predicción intra de la componente de crominancia (por ejemplo, `intra_chroma_pred_mode`) puede codificarse y generarse como el tren de bits.

Método de decodificación

55 En lo sucesivo se describirá, con referencia a la FIG. 25, un método para realizar la decodificación mediante un aparato de decodificación según una realización que utiliza el método descrito anteriormente. El aparato de decodificación según una realización puede incluir una memoria y al menos un procesador, y el al menos un procesador puede

realizar el siguiente método de decodificación.

Primero, el aparato de decodificación puede determinar un bloque actual dividiendo una imagen (S2510). Por ejemplo, el aparato de decodificación puede determinar el bloque actual dividiendo la imagen como se describió anteriormente con referencia a las FIGS. 4 a 6. En un proceso de división según una realización, se puede utilizar información de división de imagen obtenida de un tren de bits.

A continuación, el aparato de decodificación puede identificar si se aplica un modo de paleta para el bloque actual, basándose en una bandera de modo de paleta (por ejemplo, `pred_mode_plt_flag`) obtenido del tren de bits (S2520).

A continuación, el aparato de decodificación puede obtener la información de codificación de modo de paleta del bloque actual a partir del tren de bits, basándose en un tipo de árbol del bloque actual y en si el modo de paleta se aplica para el bloque actual (S2530). Por ejemplo, al determinar que se aplica el modo de paleta, el aparato de decodificación puede obtener la información de codificación de modo de paleta del tren de bits usando la sintaxis de `palette_coding()` como se describe con referencia a las FIGS. 14 a 19.

Mientras tanto, obtener la información de codificación de modo de paleta del bloque actual puede incluir obtener la información de codificación de modo de paleta de una componente de luminancia del bloque actual. Por ejemplo, cuando el tipo de árbol del bloque actual es un tipo de árbol único o un tipo de luminancia de árbol dual y el modo de paleta se aplica para el bloque actual, la información para la predicción del modo de paleta de la componente de luminancia del bloque actual se puede obtener del tren de bits.

En este caso, la información de codificación de modo de paleta de la componente de luminancia del bloque actual se puede obtener basándose en un tamaño del bloque de la componente de luminancia del bloque actual. Por ejemplo, como en la realización de la FIG. 13, la sintaxis de `palette_coding()` se puede realizar basándose en una anchura (por ejemplo, `cbWidth`) del bloque de la componente de luminancia del bloque actual y una altura (por ejemplo, `cbHeight`) del bloque de la componente de luminancia del bloque actual.

Mientras tanto, obtener la información de codificación de modo de paleta del bloque actual puede incluir además obtener la información de codificación de modo de paleta de la componente de crominancia del bloque actual. Por ejemplo, cuando se aplica el modo de paleta para el bloque actual y un tipo de árbol del bloque actual es un tipo de crominancia de árbol dual, se puede obtener información para la predicción del modo de paleta de la componente de crominancia del bloque actual a partir del tren de bits.

En este caso, la información de codificación de modo de paleta de la componente de crominancia del bloque actual se puede obtener basándose en un tamaño del bloque de la componente de crominancia del bloque actual. Por ejemplo, como en la realización de la FIG. 13, la sintaxis de `palette_coding()` se puede realizar basándose en una anchura (por ejemplo, `cbWidth/subWidthC`) del bloque de la componente de crominancia del bloque actual y una altura (por ejemplo, `cbHeight/subHeightC`) del bloque de la componente de crominancia del bloque actual. Aquí, `subWidthC` y `subHeightC` pueden ser proporciones de la altura y la anchura del bloque de la componente de luminancia con respecto al bloque de la componente de crominancia. En una realización, `subWidthC` y `subHeightC` se pueden determinar basándose en `chroma_format_idc` y `separate_colour_plane_flag` como se muestra en la FIG. 10.

A continuación, cuando no se aplica el modo de paleta para el bloque actual, el aparato de decodificación puede obtener información de predicción de la componente de crominancia del bloque actual a partir del tren de bits (S2540). Por ejemplo, cuando no se aplica el modo de paleta, el aparato de decodificación puede obtener información de predicción CCLM (por ejemplo, `cclm_mode_flag`, `cclm_mode_idx`) o información de predicción intra de la componente de crominancia (por ejemplo, `intra_chroma_pred_mode`) del tren de bits como se describe con referencia a la FIG. 23. Mientras tanto, cuando el modo de paleta no se aplica para el bloque actual, es posible que la información de predicción de la componente de crominancia no se obtenga del tren de bits.

Más específicamente, la información para la predicción CCLM puede incluir una bandera de CCLM (por ejemplo, `cclm_mode_flag`) que especifica si se realiza predicción CCLM y un índice de modo CCLM (por ejemplo, `cclm_mode_idx`) que especifica un modo de predicción CCLM. La bandera de CCLM se puede obtener del tren de bits cuando la predicción CCLM está disponible para el bloque actual. El índice de modo CCLM puede obtenerse del tren de bits cuando la bandera de CCLM especifica que se realiza la predicción CCLM.

Mientras tanto, cuando la bandera de CCLM especifica que no se realiza predicción CCLM, la información de predicción intra de la componente de crominancia (por ejemplo, `intra_chroma_pred_mode`) se puede obtener del tren de bits.

Realización de aplicación

Si bien los métodos ejemplares de la presente divulgación descritos anteriormente se representan como una serie de operaciones para mayor claridad de la descripción, no se pretende limitar el orden en el que se realizan los pasos, y los pasos se pueden realizar simultáneamente o en diferente orden según sea necesario. Para implementar el método según la presente divulgación, los pasos descritos pueden incluir además otros pasos, pueden incluir pasos restantes excepto algunos de los pasos, o pueden incluir otros pasos adicionales excepto algunos pasos.

En la presente divulgación, el aparato de codificación de imagen o el aparato de decodificación de imagen que realiza una operación (paso) predeterminada puede realizar una operación (paso) de confirmar una condición o situación de ejecución de la operación (paso) correspondiente. Por ejemplo, si se describe que se realiza una operación predeterminada cuando se cumple una condición predeterminada, el aparato de codificación de imagen o el aparato de decodificación de imagen pueden realizar la operación predeterminada después de determinar si se cumple la condición predeterminada.

Las diversas realizaciones de la presente divulgación no son una lista de todas las combinaciones posibles y están destinadas a describir aspectos representativos de la presente divulgación, y las cuestiones descritas en las diversas realizaciones se pueden aplicar de forma independiente o en combinación de dos o más.

Se pueden implementar varias realizaciones de la presente divulgación en hardware, firmware, software o una combinación de los mismos. En el caso de implementar la presente divulgación mediante hardware, la presente divulgación se puede implementar con circuitos integrados de aplicación específica (ASIC), procesadores digitales de señales (DSP), dispositivos de procesamiento digital de señales (DSPD), dispositivos lógicos programables (PLD), matrices de puertas programables en campo (FPGA), procesadores generales, controladores, microcontroladores, microprocesadores, etc.

Además, el aparato de decodificación de imagen y el aparato de codificación de imagen, a los que se aplican las realizaciones de la presente divulgación, pueden incluirse en un dispositivo de transmisión y recepción de radiodifusión multimedia, un terminal de comunicación móvil, un dispositivo de vídeo de cine en casa, un dispositivo de vídeo de cine digital, una cámara de vigilancia, un dispositivo de chat por vídeo, un dispositivo de comunicación en tiempo real tal como comunicación por vídeo, un dispositivo móvil de transmisión en flujo continuo, un soporte de almacenamiento, una videocámara, un dispositivo de provisión de servicios de vídeo por demanda (VoD), un dispositivo de vídeo OTT (vídeo over-the-top), un dispositivo de provisión de servicios de transmisión en flujo continuo por Internet, un dispositivo de vídeo tridimensional (3D), un dispositivo de vídeo de videotelefonía, un dispositivo de vídeo médico y similares, y pueden usarse para procesar señales de vídeo o señales de datos. Por ejemplo, los dispositivos de vídeo OTT pueden incluir una consola de juegos, un reproductor de Blu-ray, un televisor con acceso a Internet, un sistema de cine en casa, un teléfono inteligente, una tableta PC, una grabadora de vídeo digital (DVR) o similares.

La FIG. 26 es una vista que muestra un sistema de transmisión de contenidos en flujo continuo, al que es aplicable una realización de la presente divulgación.

Como se muestra en la FIG. 26, el sistema de transmisión de contenidos en flujo continuo, al que se aplica la realización de la presente divulgación, puede incluir en líneas generales un servidor de codificación, un servidor de transmisión en flujo continuo, un servidor web, un almacenamiento de medios, un dispositivo de usuario y un dispositivo de entrada multimedia.

El servidor de codificación comprime la entrada de contenidos desde dispositivos de entrada multimedia como un teléfono inteligente, una cámara, una videocámara, etc. en datos digitales para generar un tren de bits y transmite el tren de bits al servidor de transmisión en flujo continuo. Como otro ejemplo, cuando los dispositivos de entrada multimedia, como teléfonos inteligentes, cámaras, videocámaras, etc. generan directamente un tren de bits, se puede omitir el servidor de codificación.

El tren de bits puede generarse mediante un método de codificación de imagen o un aparato de codificación de imagen, al que se aplica la realización de la presente divulgación, y el servidor de transmisión en flujo continuo puede almacenar temporalmente el tren de bits en el proceso de transmisión o recepción del tren de bits.

El servidor de transmisión en flujo continuo transmite los datos multimedia al dispositivo de usuario basándose en la solicitud de un usuario a través del servidor web, y el servidor web sirve como medio para informar al usuario sobre un servicio. Cuando el usuario solicita un servicio deseado del servidor web, el servidor web puede entregarlo a un servidor de transmisión en flujo continuo, y el servidor de transmisión en flujo continuo puede transmitir datos multimedia al usuario. En este caso, el sistema de transmisión de contenidos en flujo continuo puede incluir un servidor de control independiente. En este caso, el servidor de control sirve para controlar un comando/respuesta entre dispositivos en el sistema de transmisión de contenidos en flujo continuo.

El servidor de transmisión en flujo continuo puede recibir contenidos desde un almacenamiento de medios y/o un servidor de codificación. Por ejemplo, cuando los contenidos se reciben desde el servidor de codificación, los contenidos pueden recibirse en tiempo real. En este caso, para proporcionar un servicio de transmisión en flujo continuo uniforme, el servidor de transmisión en flujo continuo puede almacenar el tren de bits durante un tiempo predeterminado.

Ejemplos del dispositivo de usuario pueden incluir un teléfono móvil, un teléfono inteligente, una computadora portátil, un terminal de difusión digital, un asistente digital personal (PDA), un reproductor multimedia portátil (PMP), navegación, una computadora tipo pizarra, tabletas PC, ultrabooks, dispositivos portátiles (por ejemplo, relojes inteligentes, gafas inteligentes, pantallas montadas en la cabeza), televisores digitales, computadoras de escritorio, señalización digital y similares.

Cada servidor en el sistema de transmisión de contenidos en flujo continuo puede funcionar como un servidor

distribuido, en cuyo caso los datos recibidos de cada servidor pueden distribuirse.

5 El alcance de la divulgación incluye software o comandos ejecutables por máquina (por ejemplo, un sistema operativo, una aplicación, firmware, un programa, etc.) para permitir que las operaciones de acuerdo con los métodos de diversas realizaciones se ejecuten en un aparato o una computadora, un soporte no transitorio legible por computadora que tenga dicho software o comandos almacenados en el mismo y ejecutable en el aparato o la computadora.

Aplicabilidad industrial

Las realizaciones de la presente divulgación se pueden usar para codificar o decodificar una imagen.

REIVINDICACIONES

1. Un método de decodificación de imagen realizado por un aparato de decodificación de imagen, comprendiendo el método de decodificación de imagen:
- determinar un bloque actual dividiendo una imagen;
- 5 identificar si se aplica un modo de paleta para el bloque actual, basándose en una bandera de modo de paleta obtenida de un tren de bits;
- determinar si un tipo de árbol del bloque actual es un tipo de crominancia de árbol dual y si el modo de paleta se aplica al bloque actual;
- determinar si el modo de paleta se aplica al bloque actual; y
- 10 obtener información de predicción de la componente de crominancia del bloque actual a partir del tren de bits, basándose en que el modo de paleta no se aplica al bloque actual,
- caracterizado por que la determinación de si el modo de paleta se aplica al bloque actual se realiza en función de que el tipo de árbol del bloque actual no sea el tipo de crominancia de árbol dual o que el modo de paleta no se aplique al bloque actual.
- 15 2. El método de decodificación de imagen de la reivindicación 1,
- en el que la información de predicción de la componente de crominancia no se obtiene del tren de bits, basándose en que el modo de paleta se aplica para el bloque actual.
3. El método de decodificación de imagen de la reivindicación 1, que comprende además:
- obtener información de codificación de modo de paleta de una componente de luminancia del bloque actual,
- 20 en el que la información para la predicción del modo de paleta de la componente de luminancia del bloque actual se obtiene del tren de bits, basándose en que un tipo de árbol del bloque actual es un tipo de árbol único o un tipo de luminancia de árbol dual y el modo de paleta se aplica para el bloque actual.
4. El método de decodificación de imagen de la reivindicación 3, en el que la obtención de la información de codificación de modo de paleta de la componente de luminancia del bloque actual se realiza basándose en un tamaño de un bloque de la componente de luminancia del bloque actual.
- 25 5. El método de decodificación de imagen de la reivindicación 1, que comprende además:
- obtener información de codificación de modo de paleta de una componente de crominancia del bloque actual,
- en el que la información para la predicción del modo de paleta de la componente de crominancia del bloque actual se obtiene del tren de bits, basándose en que el modo de paleta se aplica para el bloque actual y que el tipo de árbol del bloque actual es un tipo de crominancia de árbol dual.
- 30 6. El método de decodificación de imagen de la reivindicación 5,
- en el que la obtención de la información de codificación de modo de paleta de la componente de crominancia del bloque actual se realiza basándose en un tamaño de un bloque de componente de crominancia del bloque actual.
7. El método de decodificación de imagen de la reivindicación 1,
- 35 en el que la información de predicción de la componente de crominancia es información para predicción del modelo lineal de componentes cruzadas (CCLM) o información de predicción intra de la componente de crominancia.
8. El método de decodificación de imagen de la reivindicación 7,
- 40 en el que la información para predicción CCLM comprende una bandera de CCLM que especifica si se realiza la predicción CCLM y un índice de modo CCLM que especifica un modo de predicción CCLM,
- en el que la bandera de CCLM se obtiene del tren de bits, basándose en la predicción CCLM que está disponible para el bloque actual, y
- en el que el índice de modo CCLM se obtiene del tren de bits, basándose en la bandera de CCLM que especifica que se realiza la predicción CCLM.

9. El método de decodificación de imagen de la reivindicación 8,
 en el que la información de predicción intra de la componente de crominancia se obtiene del tren de bits, basándose en la bandera de CCLM que especifica que no se realiza la predicción CCLM.
- 5 10. Un método de codificación de imagen realizado por un aparato de codificación de imagen, comprendiendo el método de codificación de imagen:
- determinar un bloque actual dividiendo una imagen;
- determinar un modo de predicción del bloque actual;
- codificar una bandera de modo de paleta que especifica si el modo de predicción del bloque actual es un modo de paleta, en función de si el modo de predicción del bloque actual es el modo de paleta;
- 10 determinar si un tipo de árbol del bloque actual es un tipo de crominancia de árbol dual y si el modo de paleta se aplica al bloque actual;
- determinar si el modo de paleta se aplica al bloque actual; y
- codificar información de predicción de la componente de crominancia del bloque actual, basándose en que el modo de predicción del bloque actual no es el modo de paleta,
- 15 caracterizado por que la determinación de si el modo de paleta se aplica al bloque actual se realiza en función de que el tipo de árbol del bloque actual no sea el tipo de crominancia de árbol dual o que el modo de paleta no se aplique al bloque actual.
11. El método de codificación de imagen de la reivindicación 10, en el que la información de predicción de la componente de crominancia no está codificada, basándose en que se aplica el modo de paleta para el bloque actual.
- 20 12. El método de codificación de imagen de la reivindicación 11, en el que la información de predicción de la componente de crominancia es información para predicción del modelo lineal de componentes cruzadas (CCLM) o información de predicción intra de la componente de crominancia.
13. El método de codificación de imagen de la reivindicación 10,
- 25 en el que se codifica información de codificación de modo de paleta en la que una componente de crominancia del bloque actual está codificada en el modo de paleta, basándose en que se aplica el modo de paleta para el bloque actual y en que el tipo de árbol del bloque actual es un tipo de crominancia de árbol dual, y
- en el que se codifica información para predicción del modelo lineal de componentes cruzadas (CCLM) o información de predicción intra de la componente de crominancia como información de predicción de la componente de crominancia, basándose en que el modo de paleta no se aplica para el bloque actual.
- 30 14. Un método para transmitir un tren de bits generado por un método de codificación de imagen, comprendiendo el método de codificación de imagen:
- determinar un bloque actual dividiendo una imagen;
- determinar un modo de predicción del bloque actual;
- 35 codificar una bandera de modo de paleta que especifica si el modo de predicción del bloque actual es un modo de paleta, en función de si el modo de predicción del bloque actual es el modo de paleta;
- determinar si un tipo de árbol del bloque actual es un tipo de crominancia de árbol dual y si el modo de paleta se aplica al bloque actual;
- determinar si el modo de paleta se aplica al bloque actual; y
- 40 codificar información de predicción de la componente de crominancia del bloque actual, basándose en que el modo de predicción del bloque actual no es el modo de paleta,
- caracterizado por que la determinación de si el modo de paleta se aplica al bloque actual se realiza basándose en que el tipo de árbol del bloque actual no es el tipo de crominancia de árbol dual o que el modo de paleta no se aplica al bloque actual.

FIG. 1

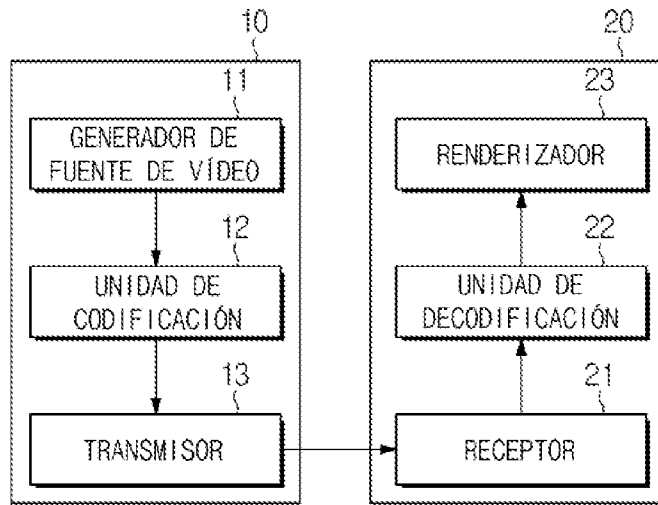


FIG. 2

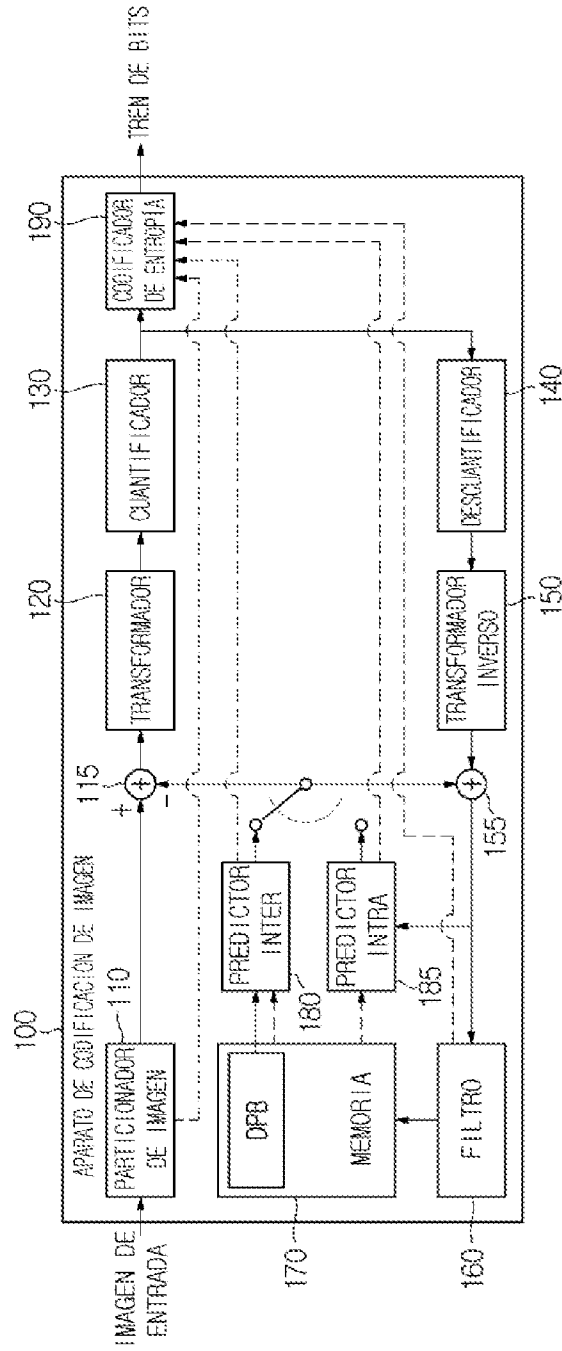


FIG. 4

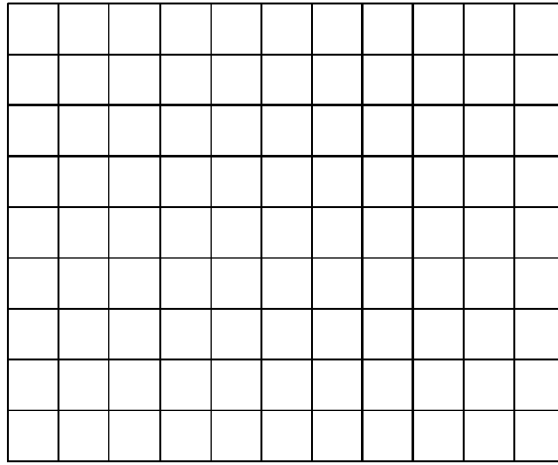


FIG. 5

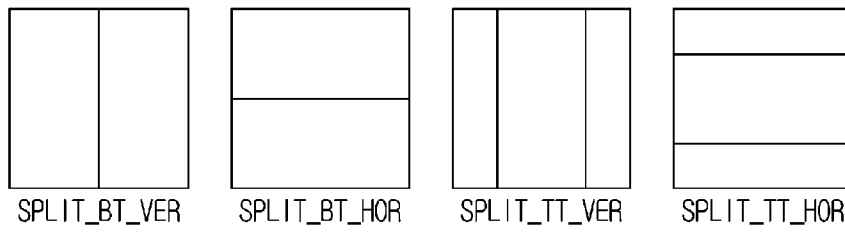


FIG. 6

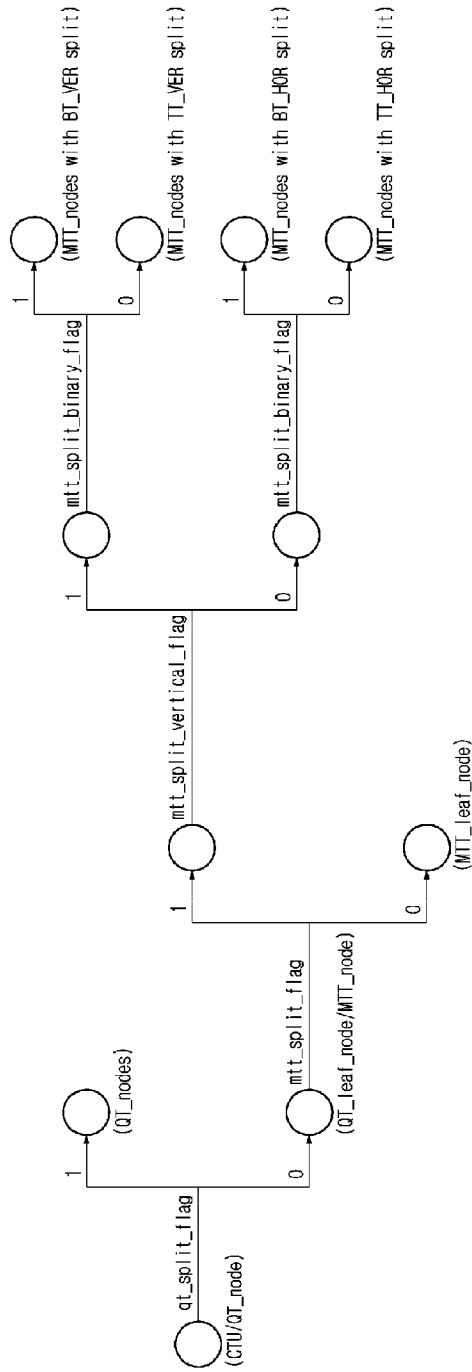


FIG. 7

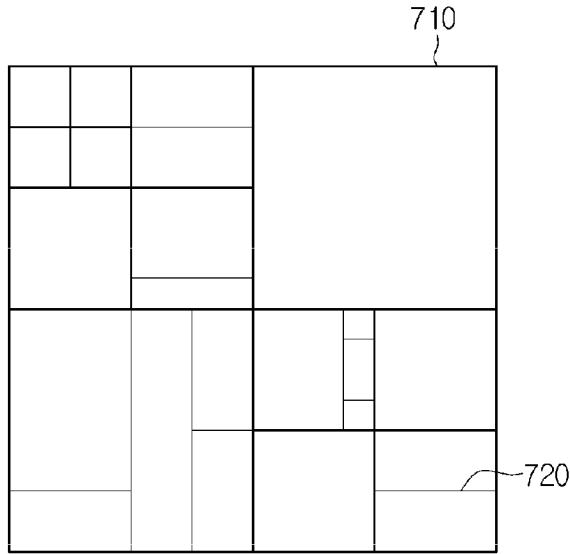


FIG. 8

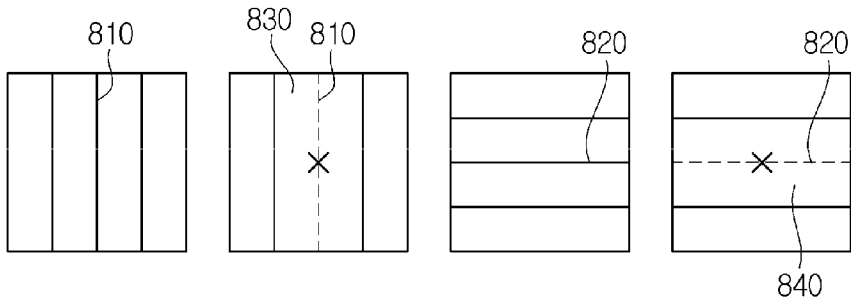


FIG. 9

	Descriptor
seq parameter set rbsp() {	
sps decoding parameter set id	u(4)
sps video parameter set id	u(4)
sps max sublayers minus 1	u(3)
sps reserved zero 5bits	u(5)
profile tier level(sps max sub layers minus 1)	
gra enabled flag	u(1)
sps seq parameter set id	ue(v)
chroma format idc	ue(v)
if(chroma format idc == 3)	
separate colour plane flag	u(1)
...	...

FIG. 10

chroma_format_idc	separate_colour_plane_flag	ChromaArrayType	Formato de crominancia	SubWidthC	SubHeightC
0	0	0	Monocromo	1	1
1	0	1	4:2:0	2	2
2	0	2	4:2:2	2	1
3	0	3	4:4:4	1	1
3	1	0	4:4:4	1	1

FIG. 11

exploración de recorrido horizontal exploración de recorrido vertical

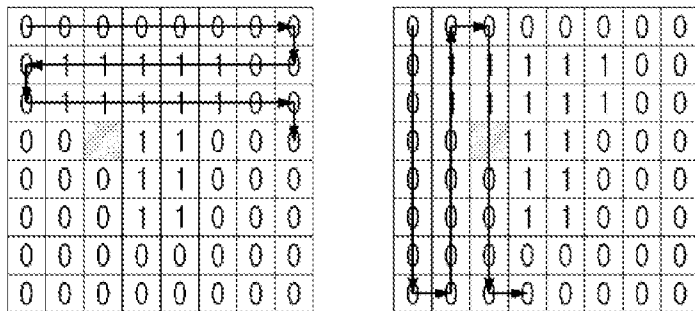


FIG. 12

	Descriptor
coding_unit(x0, y0, cbWidth, cbHeight, trecType, modeType) {	
if(slice_type != 1 sps_IBC_enabled_flag sps_plt_enabled_flag) {	
if(treeType != DUAL_TREE_CHROMA && !((cbWidth == 4 && cbHeight == 4) modeType == MODE_INTRA) && !sps_IBC_enabled_flag)	
cu_skip_flag[x0][y0]	ae(v)
if(cu_skip_flag[x0][y0] == 0 && slice_type != 1 && !(cbWidth == 4 && cbHeight == 4) && modeType == MODE_ALL)	
pred_mode_flag	ae(v)
if(((slice_type == 1 && cu_skip_flag[x0][y0] == 0) (slice_type != 1 && (CuPredMode[x0][y0] != MODE_INTRA (cbWidth == 4 && cbHeight == 4 && cu_skip_flag[x0][y0] == 0))) && cbWidth <= 64 && cbHeight <= 64) && modeType != MODE_INTER) { [Ed. (BB): This true for 4:4:4 palette as well?]	
if(sps_IBC_enabled_flag && treeType != DUAL_TREE_CHROMA)	
pred_mode_IBC_flag	ae(v)
if(pred_mode_IBC_flag[x0][y0] == 0 && sps_plt_enabled_flag)	
pred_mode_plt_flag	ae(v)
}	
}	
}	

FIG. 13

if(CuPredMode[x0][y0] == MODE_INTRA) {	
if(trecType == SINGLE_TREE trecType == DUAL_TREE_LUMA) {	
if(pred_mode_plt_flag) {	
if(trecType == DUAL_TREE_LUMA)	
palette_coding(x0, y0, cbWidth, cbHeight, 0, 1)	
else /* SINGLE_TREE */	
palette_coding(x0, y0, cbWidth, cbHeight, 0, 3)	
} else {	
if(sps_bdpcm_enabled_flag && cbWidth <= MaxTsSize && cbHeight <= MaxTsSize)	
intra_bdpcm_flag[x0][y0]	ae(v)
if(intra_bdpcm_flag[x0][y0])	
intra_bdpcm_dir_flag[x0][y0]	ae(v)
else {	
...	
}	
}	
if((treeType == SINGLE_TREE treeType == DUAL_TREE_CHROMA) && ChromaArrayType != 0) {	
if(pred_mode_plt_flag && treeType == DUAL_TREE_CHROMA)	
palette_coding(x0, y0, cbWidth / SubWidthC, cbHeight / SubHeightC, 1, 2)	
else {	
if(CclmEnabled)	
eclm_mode_flag	ae(v)
if(eclm_mode_flag)	
eclm_mode_idx	ae(v)
else	
intra_chroma_pred_mode	ae(v)
}	
}	
} else if(trecType != DUAL_TREE_CHROMA) { /* MODE_INTER or MODE_IBC */	
...	

FIG. 14

	Descriptor
palette_coding(x0, y0, cbWidth, cbHeight, startComp, numComps) {	
/* Parsing palette reuse flags */	
palettePredictionFinished = 0	
NumPredictedPaletteEntries = 0	
for(predictorEntryIdx = 0; predictorEntryIdx < PredictorPaletteSize[startComp] && !palettePredictionFinished && NumPredictedPaletteEntries[startComp] < palette_max_size; predictorEntryIdx++) {	
palette_predictor_run	ae(v)
if(palette_predictor_run != 1) {	
if(palette_predictor_run > 1)	
predictorEntryIdx += palette_predictor_run - 1	
PalettePredictorEntryReuseFlags[predictorEntryIdx] = 1	
NumPredictedPaletteEntries++	
} else	
palettePredictionFinished = 1	
}	
/* Parsing number of new palette entries */	
num_signalled_palette_entries	ae(v)
[Ed. (XX): In HEVC v4, this syntax is conditionally signaled by “if(NumPredictedPaletteEntries < palette_max_size)”]	
/* Parsing new palette entries */	
for(cIdx = startComp; cIdx < (startComp + numComps); cIdx++)	
for(i = 0; i < num_signalled_palette_entries; i++)	
new_palette_entries[cIdx][i]	ae(v)

FIG. 15

/* Parsing escape flag */	
if(CurrentPaletteSize[startComp] > 0)	
palette_escape_val_present_flag	ae(v)
/* Parsing palette indices following scanning order */	
if(MaxPaletteIndex > 1) {	
num_palette_indices_minus1	ae(v)
for(i = 0; i <= num_palette_indices_minus1; i++) {	
[Ed. (BB): In HEVC v4 SCC, there is an adjust variable used to condition the parsing of palette_idx_idc: if(MaxPaletteIndex - adjust > 0, above it is conditioned on > 0 and see also my comment in the semantics on the derivation of MaxPaletteIndex)]	
palette_idx_idc	ae(v)
PaletteIndexIdc [i] = palette_idx_idc	
}	
copy_above_indices_for_final_run_flag	ae(v)
palette_transpose_flag	ae(v)
}	
remainingNumIndices = num_palette_indices_minus1 + 1	
[Ed. (XX): In HEVC v4, delta_qp() and chroma_qp_offset() were signaled before this variable)”]	

FIG. 16

PaletteScanPos = 0	
while(PaletteScanPos < cbWidth*cbHeight) {	
xC = TraverseScanOrder[cbWidth][cbHeight][PaletteScanPos][0]	
yC = TraverseScanOrder[cbWidth][cbHeight][PaletteScanPos][1]	
[Ed. (BB): TraverseScanOrder is not defined, in HEVC v4 SCC x0 + ScanOrder[][][][] is used and that array also exist in VVC draft, use or extend this and verify that it is usewd correctly]	
if(PaletteScanPos > 0) {	
xcPrev = TraverseScanOrder[cbWidth][cbHeight][PaletteScanPos - 1][0]	
ycPrev = TraverseScanOrder[cbWidth][cbHeight][PaletteScanPos - 1][1]	
}	
CopyAboveIndicesFlag[xC][yC] = 0	
/* Parsing run types */	
if(MaxPaletteIndex > 1)	
if((!palette_transpose_flag && yC > 0) (palette_transpose_flag && xC > 0))	
&& CopyAboveIndicesFlag[xcPrev][ycPrev] == 0)	
[Ed. (XX): the condition here is different from what HEVC v4 writes]	
if(remainingNumIndices > 0 && PaletteScanPos < cbWidth*cbHeight - 1) {	
copy above palette indices flag	ae(v)
CopyAboveIndicesFlag[xC][yC] = copy_above_palette_indices_flag	
} else {	
if(PaletteScanPos == cbWidth*cbHeight - 1 && remainingNumIndices > 0)	
CopyAboveIndicesFlag[xC][yC] = 0	
else	
CopyAboveIndicesFlag[xC][yC] = 1	
}	

FIG. 17

/* Palette Index adjustment */	
if(CopyAboveIndicesFlag[xC][yC] == 0) {	
currNumIndices - num_palette_indices_minus1 + 1 - remainingNumIndices	
PaletteIndexMap[xC][yC] = PaletteIndexIdx[currNumIndices]	
RefLevel = MaxPaletteIndex	
if(PaletteScanPos > 0) {	
if(CopyAboveIndicesFlag[xcPrev][ycPrev] == 0) {	
RefLevel = PaletteIndexMap[xcPrev][ycPrev]	
} else {	
if(!palette_transpose_flag)	
RefLevel = PaletteIndexMap[xC][yC - 1]	
else	
RefLevel = PaletteIndexMap[xC - 1][yC]	
}	
}	
if(PaletteIndexIdx[currNumIndices] >= RefLevel)	
PaletteIndexMap[xC][yC]++	
[Ed. (XX): this part is different from what HEVC v4 writes]	
}	

FIG. 18

/* Parsing run length */	
PaletteRun = cbWidth * cbHeight - PaletteScanPos	
[Ed. (XX): the condition here is different from what HEVC v4, where "PaletteRun = nCbS * nCbS - PaletteScanPos - 1"]	
if(MaxPaletteIndex > 1) {	
if(CopyAboveIndicesFlag[xC][yC] == 0)	
remainingNumIndices -= 1	
PaletteMaxRun = nCbS * nCbS - PaletteScanPos - 1 - remainingNumIndices	
copy above indices for final run flag	
if(remainingNumIndices > 0 CopyAboveIndicesFlag[xC][yC] !=	
copy above indices for final run flag)	
if(PaletteMaxRun > 0) {	
palette_run_prefix	ae(v)
if((palette_run_prefix > 1) && (PaletteMaxRun !=	
(1 << (palette_run_prefix - 1)))	
palette_run_suffix	ae(v)
} else {	
PaletteRun = cbWidth * cbHeight - PaletteScanPos	
}	
for(runPos = 1; runPos < PaletteRun; runPos++) {	
xR = TraverseScanOrder[cbWidth][cbHeight][PaletteScanPos + runPos][0]	
yR = TraverseScanOrder[cbWidth][cbHeight][PaletteScanPos + runPos][1]	
if(CopyAboveIndicesFlag[xC][yC] == 0) {	
CopyAboveIndicesFlag[xR][yR] = 0	
PaletteIndexMap[xR][yR] = PaletteIndexMap[xC][yC]	
} else {	
CopyAboveIndicesFlag[xR][yR] = 1	
if(!palette_transpose_flag)	
PaletteIndexMap[xR][yR] = PaletteIndexMap[xR][yR - 1]	
[Ed. (XX): the condition here is different from what HEVC v4, where assignment of PaletteIndexMap[xR][yR] does not depend on palette_transpose_flag]	
else	
PaletteIndexMap[xR][yR] = PaletteIndexMap[xR - 1][yR]	
}	
}	
PaletteScanPos += PaletteRun	
}	

FIG. 19

/* Parsing escape values */	
if(palette_escape_val_present_flag) {	
for(cIdx = startComp; cIdx < (startComp + numComps); cIdx++)	
for(sPos = 0; sPos < cbWidth * cbHeight; sPos++) {	
xC = TraverseScanOrder[cbWidth][cbHeight][sPos][0]	
yC = TraverseScanOrder[cbWidth][cbHeight][sPos][1]	
if(PaletteIndexMap[cIdx][xC][yC] == (MaxPaletteIndex - 1)) {	
palette_escape_val	ae(v)
PaletteEscapeVal[cIdx][xC][yC] = palette_escape_val	
}	
}	
}	

FIG. 20

```

numPredictedPaletteEntries = 0
for( i = 0; i < PredictorPaletteSize[ startComp ]; i++)
    if( PalettePredictorEntryReuseFlags[ i ] ) {
        for( cIdx = startComp; cIdx < ( startComp + numComps ); cIdx++)
            CurrentPaletteEntries[ cIdx ][ numPredictedPaletteEntries ] = PredictorPaletteEntries[ cIdx ][ i ]
            numPredictedPaletteEntries++
    }

for( cIdx = startComp; cIdx < (startComp + numComps); cIdx++)
    for( i = 0; i < num_signalled_palette_entries[startComp]; i-- )
        CurrentPaletteEntries[ cIdx ][ numPredictedPaletteEntries + i ] = new_palette_entries[ cIdx ][ i ]
    
```

FIG. 21

	if((treeType == SINGLE_TREE treeType == DUAL_TREE_CHROMA) && ChromaArrayType != 0) {	
	if(pred_mode_plt_flag && treeType == DUAL_TREE_CHROMA)	
	palette_coding(x0, y0, cbWidth / SubWidthC, cbHeight / SubHeightC, 1, 2)	
2110	else if (!pred_mode_plt_flag) {	
	if(CclmEnabled)	
	cclm_mode_flag	ae(v)
	if(cclm_mode_flag)	
2120	cclm_mode_idx	ae(v)
	else	
	intra_chroma_pred_mode	ac(v)
	}	
	}	

FIG. 22

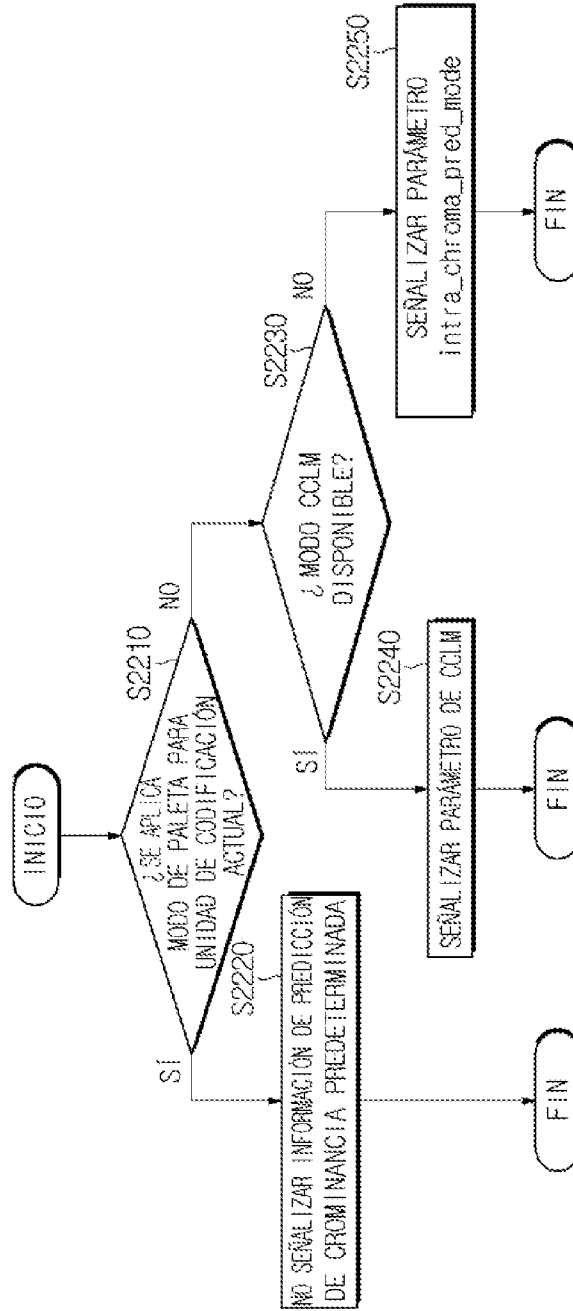


FIG. 23

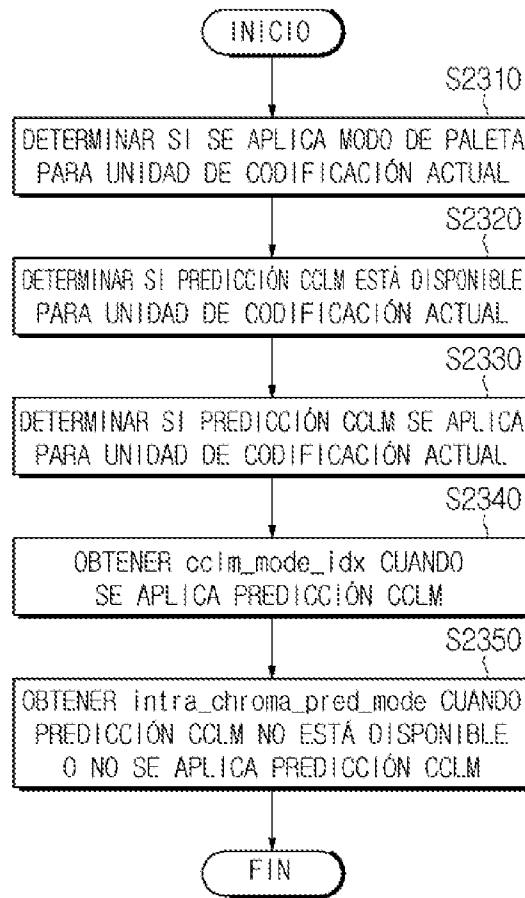


FIG. 24

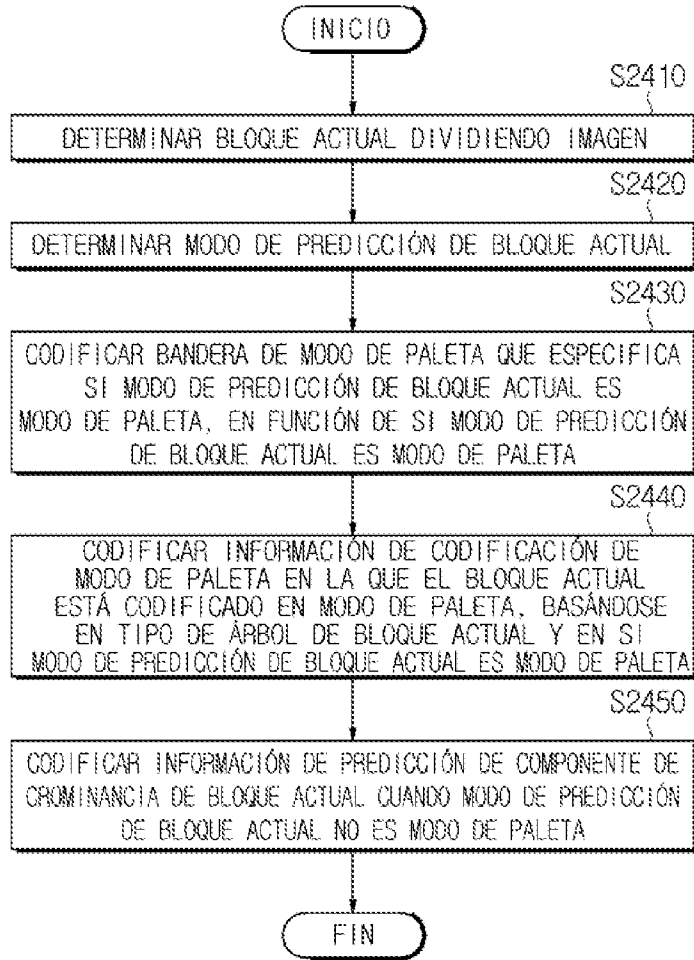


FIG. 25

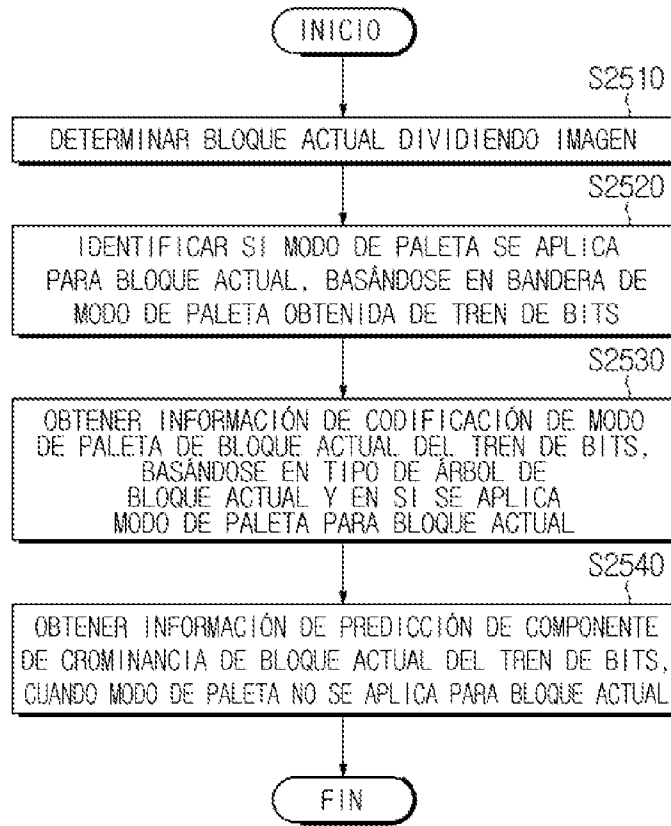


FIG. 26

