

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **3 025 935**

51 Int. Cl.:

H04N 19/124 (2014.01)

H04N 19/157 (2014.01)

H04N 19/176 (2014.01)

H04N 19/61 (2014.01)

H04N 19/70 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **05.10.2020 PCT/KR2020/013495**

87 Fecha y número de publicación internacional: **08.04.2021 WO21066618**

96 Fecha de presentación y número de la solicitud europea: **05.10.2020 E 20872437 (7)**

97 Fecha y número de publicación de la concesión europea: **16.04.2025 EP 4040792**

54 Título: **Codificación de imagen o vídeo basada en la señalización de información relacionada con la codificación de paletas y de salto de transformada**

30 Prioridad:
05.10.2019 US 201962911221 P

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
10.06.2025

73 Titular/es:
**LG ELECTRONICS INC. (100.00%)
128, Yeoui-daero, Yeongdeungpo-gu
SEOUL 07336, KR**

72 Inventor/es:
**YOO, SUNMI;
CHOI, JUNGHAH y
CHOI, JANGWON**

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

ES 3 025 935 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

Codificación de imagen o vídeo basada en la señalización de información relacionada con la codificación de paletas y de salto de transformada

5

Antecedentes de la descripción

Campo de la descripción

10 La presente tecnología se refiere a la codificación de vídeo o imagen y, por ejemplo, a una tecnología de codificación de imagen o vídeo basada en la señalización de información relacionada con el salto de transformada y la codificación de paleta.

Técnica relacionada

15

Recientemente, la demanda de imagen/vídeo de alta resolución y alta calidad como, por ejemplo, imagen/vídeo de 4K, 8K o más de definición ultra alta (UHD, por sus siglas en inglés), está aumentando en diversos campos. A medida que la resolución o calidad de imagen/vídeo se hace más alta, se transmite relativamente más cantidad de información o bits que para datos de imagen/vídeo convencionales. Por lo tanto, si los datos de imagen/vídeo se transmiten a través de un medio como, por ejemplo, una línea de banda ancha cableada/inalámbrica existente o se almacenan en un medio de almacenamiento heredado, los costes de transmisión y almacenamiento se aumentan fácilmente.

20

Además, están creciendo intereses y demandas de contenidos de realidad virtual (VR, por sus siglas en inglés) y realidad artificial (AR, por sus siglas en inglés), y medios inmersivos como, por ejemplo, hologramas; y también están creciendo la difusión de imágenes/vídeos que presentan características de imagen/vídeo diferentes de las de una imagen/vídeo real como, por ejemplo, imágenes/vídeos de juegos.

25

Por lo tanto, se requiere una técnica de compresión de imagen/vídeo altamente eficiente para comprimir y transmitir, almacenar o reproducir imágenes/vídeos de alta calidad y alta resolución que muestran diversas características como se ha descrito anteriormente.

30

Además, existe la necesidad de un esquema para mejorar la eficiencia global de codificación de imagen/vídeo clasificándose eficientemente si codificar información relacionada según la dependencia y no dependencia de información que es absolutamente necesaria o auxiliar usada en la realización de salto de transformada y codificación de paleta. El documento de BROSS B Y OTROS: "Versatile Video Coding (Draft 6)", 15. Reunión de JVET; 20190703 - 20190712; GOTHENBURG; (EL EQUIPO DE EXPLORACIÓN DE VÍDEO CONJUNTO DE ISO/IEC JTC1/SC29/WG11 E ITU-T SG.16), n.º JVET-O2001-vE; JVET-O2001 31 de julio de 2019 (31-07-2019), páginas 1-455, XP030293944, recuperado de Internet: URL: https://jvet-experts.org/doc_end_user/documents/15_Gothenburg/wg11/JVET-O2001-v14.zip JVET-O2001-vE.docx [recuperado el 31-07-2019] describe señalización condicional, p. ej., sintaxis de señalización para habilitar BDPCM basándose en información habilitada de salto de transformada en el nivel de SPS. No describe la señalización de la información de parámetro de cuantificación mínimo basándose en una condición de que el valor de la información habilitada de salto de transformada sea igual a 1 y una condición de que el valor de la información habilitada de codificación de paleta sea igual a 1.

35

40

45

Compendio

Un objeto técnico del presente documento es proveer un método y un aparato para mejorar la eficiencia de codificación de vídeo/imagen.

50

Otro objeto técnico del presente documento es proveer un método y un aparato para analizar/señalizar de manera eficiente información relacionada con el salto de transformada y/o la codificación de paleta.

55

Incluso otro objeto técnico más del presente documento es proveer un método y un aparato para determinar de manera eficiente si llevar a cabo la codificación según la dependencia y/o no dependencia de la información que se usa durante el salto de transformada y/o la codificación de paleta.

60

Incluso otro objeto técnico más del presente documento es proveer un método y un aparato para definir una condición dependiente para analizar eficazmente un elemento sintáctico que tiene dependencia con respecto a un elemento sintáctico de alto nivel relacionado con salto de transformada y/o codificación de paleta, y determinar si llevar a cabo el análisis basándose en la condición dependiente.

65

Según una realización del presente documento, la información habilitada para el salto de transformada y la información habilitada para la paleta pueden señalizarse a través de un conjunto de parámetros de secuencia (SPS, por sus siglas en inglés), y si analizar/señalizar la información de parámetro de cuantificación mínimo

relacionada con el parámetro de cuantificación mínimo permitido para un modo de salto de transformada puede determinarse basándose en al menos una de la información habilitada para el salto de transformada y la información habilitada para la paleta. Por ejemplo, la información de parámetro de cuantificación mínimo puede analizarse/señalizarse a través del SPS basándose en una condición de que un valor de la información habilitada de salto de transformada sea 1 o un valor de la información habilitada de codificación de paleta sea 1.

Según una realización del presente documento, se provee un método de decodificación de vídeo/imagen llevado a cabo por un aparato de decodificación. El método de decodificación de vídeo/imagen puede incluir un método descrito en realizaciones del presente documento.

Según una realización del presente documento, se provee un aparato de decodificación que lleva a cabo la decodificación de vídeo/imagen. El aparato de decodificación puede llevar a cabo un método descrito en las realizaciones del presente documento.

Según una realización del presente documento, se provee un método de codificación de vídeo/imagen llevado a cabo por un aparato de codificación. El método de codificación de vídeo/imagen puede incluir un método descrito en las realizaciones del presente documento.

Según una realización del presente documento, se provee un aparato de codificación que lleva a cabo la codificación de vídeo/imagen. El aparato de codificación puede llevar a cabo un método descrito en realizaciones del presente documento.

Según una realización del presente documento, se provee un medio de almacenamiento digital legible por ordenador que almacena información de vídeo/imagen codificada generada según un método de codificación de vídeo/imagen descrito en al menos una de las realizaciones del presente documento.

Según una realización del presente documento, se provee un medio de almacenamiento digital legible por ordenador que almacena información codificada o información de vídeo/imagen codificada que hace que un aparato de decodificación lleve a cabo un método de decodificación de vídeo/imagen descrito en al menos una de las realizaciones del presente documento.

El presente documento puede tener varios efectos. Por ejemplo, según una realización del presente documento, se puede mejorar la eficiencia global de compresión de imagen/vídeo. Además, según una realización del presente documento, la información relacionada con el salto de transformada y/o la codificación de paleta puede analizarse/señalizarse de manera eficiente. Además, según una realización del presente documento, si llevar a cabo la codificación puede determinarse de manera efectiva según la dependencia y/o no dependencia de la información que se usa durante el salto de transformada y/o la codificación de paleta. Además, según una realización del presente documento, es posible una codificación eficiente definiendo la condición dependiente para analizar eficazmente el elemento sintáctico que tiene dependencia con respecto al elemento sintáctico de alto nivel relacionado con el salto de transformada y/o la codificación de paleta, y determinando si llevar a cabo el análisis sintáctico según la condición dependiente. Además, según una realización del presente documento, los bits que se transmiten pueden guardarse determinando si llevar a cabo un análisis sintáctico según la condición dependiente con respecto al elemento sintáctico de alto nivel relacionado con el salto de transformada y/o la codificación de paleta.

Dichas realizaciones están destinadas a ilustrar la invención y no a limitar su alcance: el alcance de la invención está definido por las reivindicaciones anexas.

La invención se describe en las figuras 10, 12 junto con su descripción asociada. Algunas veces, las expresiones "realización" e "invención" se usan también con referencia a otras enseñanzas incluidas en la descripción. En estos casos, se deben interpretar como unas que significan " ejemplo(s) adicional(es) que no representa(n) la invención", a menos que estén relacionadas con el objeto que cae dentro del alcance de las reivindicaciones anexas.

Breve descripción de los dibujos

La FIG. 1 ilustra esquemáticamente un ejemplo de un sistema de codificación de vídeo/imagen al que son aplicables realizaciones del presente documento.

La FIG. 2 es un diagrama que ilustra esquemáticamente la configuración de un aparato de codificación de vídeo/imagen al que son aplicables realizaciones del presente documento.

La FIG. 3 es un diagrama que explica esquemáticamente la configuración de un aparato de decodificación de vídeo/imagen al que son aplicables realizaciones del presente documento.

La FIG. 4 ilustra un ejemplo de un método de codificación de vídeo/imagen esquemático al que son aplicables

realizaciones del presente documento.

La FIG. 5 ilustra un ejemplo de un método de decodificación de vídeo/imagen esquemático al que son aplicables realizaciones del presente documento.

5

La FIG. 6 ilustra esquemáticamente un ejemplo de un método de codificación de entropía al que son aplicables realizaciones del presente documento, y la FIG. 7 ilustra esquemáticamente un codificador de entropía en un aparato de codificación.

10

La FIG. 8 ilustra esquemáticamente un ejemplo de un método de decodificación de entropía al que son aplicables realizaciones del presente documento, y la FIG. 9 ilustra esquemáticamente un decodificador de entropía en un aparato de decodificación.

15

Las FIGS. 10 y 11 ilustran esquemáticamente un método de codificación de vídeo/imagen y un ejemplo de componentes relacionados según la(s) realización(es) del presente documento.

Las FIGS. 12 y 13 ilustran esquemáticamente un método de decodificación de vídeo/imagen y un ejemplo de componentes relacionados según la(s) realización(es) del presente documento.

20

La FIG. 14 ilustra un ejemplo de un sistema de transmisión continua de contenido al que son aplicables las realizaciones descritas en el presente documento.

Descripción de realizaciones a modo de ejemplo

25

Los términos comúnmente usados en esta memoria descriptiva se usan para describir una realización específica y no se usan para limitar el espíritu técnico del presente documento. Una expresión del número singular incluye expresiones plurales a menos que se exprese evidentemente lo contrario en el contexto. Un término como, por ejemplo, "incluir" o "tener" en esta memoria descriptiva, debe entenderse que indica la existencia de una característica, número, etapa, operación, elemento, parte o una combinación de los mismos descritos en la memoria descriptiva y no excluye la existencia o la posibilidad de la adición de una o más características, números, etapas, operaciones, elementos, partes o una combinación diferentes de los mismos.

30

Mientras tanto, los elementos en los dibujos descritos en el presente documento se ilustran independientemente en aras de la descripción relacionada con diferentes funciones características. Esto no significa que cada uno de los elementos se implemente como hardware separado o software separado. Por ejemplo, al menos dos de los elementos pueden combinarse para formar un único elemento, o un único elemento puede dividirse en múltiples elementos. Una realización en la que los elementos se combinan y/o separan también se incluye en el alcance de los derechos del presente documento a menos que se desvíe de la esencia del presente documento.

35

40

En el presente documento, el término "A o B" puede significar "solo A", "solo B" o "tanto A como B". En otras palabras, en el presente documento, el término "A o B" puede interpretarse para indicar "A y/o B". Por ejemplo, en el presente documento, el término "A, B o C" puede significar "solo A", "solo B", "solo C" o "cualquier combinación de A, B y C".

45

Una barra inclinada "/" o una coma usada en el presente documento puede significar "y/o". Por ejemplo, "A/B" puede significar "A y/o B". Por consiguiente, "A/B" puede significar "solo A", "solo B" o "tanto A como B". Por ejemplo, "A, B, C" puede significar "A, B o C".

50

En el presente documento, "al menos uno de A y B" puede significar "solo A", "solo B" o "tanto A como B". Además, en el presente documento, la expresión "al menos uno de A o B" o "al menos uno de A y/o B" puede interpretarse igual que "al menos uno de A y B".

55

Además, en el presente documento, "al menos uno de A, B y C" puede significar "solo A", "solo B", "solo C" o "cualquier combinación de A, B y C". Además, "al menos uno de A, B o C" o "al menos uno de A, B y/o C" puede significar "al menos uno de A, B y C".

60

Además, los paréntesis usados en el presente documento pueden significar "por ejemplo". Específicamente, en el caso de que se exprese "predicción (intrapredicción)", puede indicarse que se propone "intrapredicción" como ejemplo de "predicción". En otras palabras, el término "predicción" en el presente documento no se limita a "intrapredicción", y puede indicarse que "intrapredicción" se propone como un ejemplo de "predicción". Además, incluso en el caso de que se exprese "predicción (es decir, intrapredicción)", puede indicarse que "intrapredicción" se propone como un ejemplo de "predicción".

65

El presente documento se refiere a la codificación de vídeo/imagen. Por ejemplo, los métodos/realizaciones descritos en el presente documento se aplican a un método descrito en el estándar de codificación de vídeo versátil (VVC, por sus siglas en inglés). Además, los métodos/realizaciones descritos en el presente documento

pueden aplicarse a un método descrito en el estándar de codificación de vídeo esencial (EVC, por sus siglas en inglés), el estándar de Vídeo de AOMedia 1 (AV1), el estándar de codificación de vídeo de audio de 2.ª generación (AVS2) o el estándar de codificación de vídeo/imagen de próxima generación (p. ej., H.267 o H.268, etc.).

5

El presente documento presenta diversas realizaciones de codificación de vídeo/imagen, y las realizaciones pueden llevarse a cabo en combinación entre sí a menos que se describa lo contrario.

10

En el presente documento, un vídeo puede significar un conjunto de una serie de imágenes según el paso del tiempo. Una imagen significa generalmente una unidad que representa una imagen en un período específico, y un segmento/mosaico es una unidad que constituye una parte de la imagen en codificación. El segmento/mosaico puede incluir una o más unidades de árbol de codificación (CTU, por sus siglas en inglés). Una imagen puede consistir en uno o más segmentos/mosaicos. Un mosaico es una región rectangular de CTU dentro de una columna de mosaicos particular y una fila de mosaicos particular en una imagen. La columna de mosaicos es una región rectangular de CTU que tiene una altura igual a la altura de la imagen y un ancho especificado por los elementos sintácticos en el conjunto de parámetros de imagen. La fila de mosaicos es una región rectangular de CTU que tiene un ancho especificado por los elementos sintácticos en el conjunto de parámetros de imagen y una altura igual a la altura de la imagen. Un escaneo de mosaicos es un ordenamiento secuencial específico de CTU que divide una imagen en la que las CTU se ordenan consecutivamente en un escaneo de trama de CTU en un mosaico, mientras que los mosaicos en una imagen se ordenan consecutivamente en un escaneo de trama de los mosaicos de la imagen. Un segmento incluye un número entero de mosaicos completos o un número entero de filas de CTU completas consecutivas dentro de un mosaico de una imagen que puede estar contenida exclusivamente en una única unidad NAL.

25

Mientras tanto, una imagen puede dividirse en dos o más subimágenes. La subimagen puede ser una región rectangular de uno o más segmentos dentro de la imagen.

30

Un píxel o un pel puede significar la unidad más pequeña que constituye una imagen. Asimismo, 'muestra' puede usarse como un término correspondiente a un píxel. Una muestra puede representar generalmente un píxel o un valor de un píxel, y puede representar solo un valor de píxel/píxel de un componente de luma o solo un valor de píxel/píxel de un componente de croma. Alternativamente, una muestra puede significar un valor de píxel en el dominio espacial, o puede significar un coeficiente de transformada en el dominio de frecuencia cuando el valor de píxel se transforma en el dominio de frecuencia.

35

Una unidad puede representar una unidad básica de procesamiento de imágenes. La unidad puede incluir al menos una de una región específica de la imagen e información relacionada con la región. Una unidad puede incluir un bloque de luma y dos bloques de croma (p. ej., cb, cr). La unidad puede usarse indistintamente con términos como, por ejemplo, bloque o área en algunos casos. En un caso general, un bloque MxN puede incluir muestras (o matrices de muestras) o un conjunto (o matriz) de coeficientes de transformada de M columnas y N filas.

40

Asimismo, en el presente documento, puede omitirse al menos una de cuantificación/descuantificación y/o transformada/transformada inversa. Cuando se omite la cuantificación/descuantificación, el coeficiente de transformada cuantificado puede denominarse coeficiente de transformada. Cuando se omite la transformada/transformada inversa, los coeficientes de transformada pueden denominarse coeficientes o coeficientes residuales, o aún pueden denominarse coeficientes de transformada en aras de la uniformidad de la expresión.

45

50

En el presente documento, un coeficiente de transformada cuantificado y un coeficiente de transformada pueden denominarse un coeficiente de transformada y un coeficiente de transformada escalado, respectivamente. En este caso, la información residual puede incluir información sobre el(los) coeficiente(s) de transformada, y la información sobre el(los) coeficiente(s) de transformada puede señalizarse a través de una sintaxis de codificación residual. Los coeficientes de transformada pueden derivarse basándose en información residual (o información sobre coeficiente(s) de transformada), y los coeficientes de transformada escalados pueden derivarse a través de transformada inversa (escalado) en los coeficientes de transformada. Las muestras residuales pueden derivarse en base a una transformada inversa (transformada) para los coeficientes de transformada escalados. Esto puede aplicarse/expresarse también en otras partes del presente documento.

55

60

En el presente documento, las características técnicas explicadas individualmente en un dibujo pueden implementarse individualmente o pueden implementarse simultáneamente.

65

A continuación, se describen realizaciones preferidas del presente documento más específicamente con referencia a los dibujos anexos. En lo sucesivo, en los dibujos, se usa el mismo numeral de referencia en el mismo elemento, y se puede omitir una descripción redundante del mismo elemento. La FIG. 1 ilustra un ejemplo de un sistema de codificación de vídeo/imagen al que pueden aplicarse las realizaciones del presente documento.

Con referencia a la FIG. 1, un sistema de codificación de vídeo/imagen puede incluir un dispositivo de origen y un dispositivo de recepción. El dispositivo de origen puede transmitir información o datos de vídeo/imagen codificados al dispositivo de recepción a través de un medio o red de almacenamiento digital en forma de un archivo o transmisión continua.

5

El dispositivo de origen puede incluir una fuente de vídeo, un aparato de codificación y un transmisor. El dispositivo receptor puede incluir un receptor, un aparato de decodificación y un renderizador. El aparato de codificación puede denominarse aparato de codificación de vídeo/imagen, y el aparato de decodificación puede denominarse aparato de decodificación de vídeo/imagen. El transmisor puede incluirse en el aparato de codificación. El receptor puede estar incluido en el aparato de decodificación. El renderizador puede incluir una pantalla, y la pantalla puede configurarse como un dispositivo separado o un componente externo.

10

La fuente de vídeo puede adquirir vídeo/imagen a través de un proceso de captura, síntesis o generación del vídeo/imagen. La fuente de vídeo puede incluir un dispositivo de captura de vídeo/imagen y/o un dispositivo generador de vídeo/imagen. El dispositivo de captura de vídeo/imagen puede incluir, por ejemplo, una o más cámaras, archivos de vídeo/imagen que incluyen vídeo/imágenes capturadas previamente y similares. El dispositivo generador de vídeo/imagen puede incluir, por ejemplo, ordenadores, tabletas y teléfonos inteligentes, y puede generar (electrónicamente) vídeo/imágenes. Por ejemplo, se puede generar un vídeo/imagen virtual a través de un ordenador o similar. En este caso, el proceso de captura de vídeo/imagen puede ser sustituido por un proceso de generación de datos relacionados.

15

20

El aparato de codificación puede codificar vídeo/imagen de entrada. El aparato de codificación puede llevar a cabo una serie de procedimientos como, por ejemplo, predicción, transformada y cuantificación para la eficiencia de compactación y codificación. Los datos codificados (información de vídeo/imagen codificada) pueden emitirse en forma de un flujo de bits.

25

El transmisor puede transmitir la imagen/información de imagen codificada o datos emitidos en forma de un flujo de bits al receptor del dispositivo receptor a través de un medio de almacenamiento digital o una red en forma de un archivo o transmisión continua. El medio de almacenamiento digital puede incluir diversos medios de almacenamiento como, por ejemplo, USB, SD, CD, DVD, Blu-ray, HDD, SSD y similares. El transmisor 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 puede recibir/extraer el flujo de bits y transmitir el flujo de bits recibido al aparato de decodificación.

30

El aparato de decodificación puede decodificar el vídeo/imagen llevando a cabo una serie de procedimientos como, por ejemplo, descuantificación, transformada inversa y predicción correspondientes al funcionamiento del aparato de codificación.

35

El renderizador puede renderizar el vídeo/imagen decodificada. El vídeo/imagen renderizada puede mostrarse a través de la pantalla.

40

La FIG. 2 es un diagrama que ilustra esquemáticamente una configuración de un aparato de codificación de vídeo/imagen al que pueden aplicarse las realizaciones del presente documento. En lo sucesivo, lo que se denomina aparato de codificación puede incluir un aparato de codificación de imágenes y/o un aparato de codificación de vídeo.

45

Con referencia a la FIG. 2, el aparato 200 de codificación puede incluir y configurarse con un particionador 210 de imágenes, un predictor 220, un procesador 230 residual, un codificador 240 de entropía, un sumador 250, un filtro 260 y una memoria 270. El predictor 220 puede incluir un interpredictor 221 y un intrapredictor 222. El procesador 230 residual puede incluir un transformador 232, un cuantificador 233, un descuantificador 234 y un transformador 235 inverso. El procesador 230 residual puede incluir además un restador 231. El sumador 250 puede denominarse reconstructor o generador de bloques reconstruidos. El particionador 210 de imágenes, el predictor 220, el procesador 230 residual, el codificador 240 de entropía, el sumador 250 y el filtro 260, que se han descrito anteriormente, pueden configurarse mediante uno o más componentes de hardware (p. ej., conjuntos de chips o procesadores de codificador) según una realización. Además, la memoria 270 puede incluir un búfer de imágenes decodificadas (DPB, por sus siglas en inglés) y también puede configurarse mediante un medio de almacenamiento digital. El componente de hardware puede incluir además la memoria 270 como un componente interno/externo.

50

55

El particionador 210 de imágenes puede dividir una imagen (o fotograma) de entrada ingresada en el aparato 200 de codificación en una o más unidades de procesamiento. Como ejemplo, la unidad de procesamiento puede denominarse unidad de codificación (CU, por sus siglas en inglés). En este caso, la unidad de codificación puede dividirse recursivamente según una estructura de árbol cuaternario árbol binario árbol ternario (QTBTTT, por sus siglas en inglés) de una unidad de árbol de codificación (CTU) o la unidad de codificación más grande (LCU, por sus siglas en inglés). Por ejemplo, una unidad de codificación puede dividirse en múltiples unidades de codificación de una profundidad más profunda basándose en una estructura

60

65

de árbol cuaternario, una estructura de árbol binario y/o una estructura de árbol ternario. En este caso, por ejemplo, se aplica primero la estructura de árbol cuaternario y la estructura de árbol binario y/o la estructura de árbol ternario pueden aplicarse más tarde. Alternativamente, la estructura de árbol binario también puede aplicarse en primer lugar. Un procedimiento de codificación según el presente documento puede llevarse a cabo basándose en una unidad de codificación final que ya no se divide. En este caso, basándose en la eficiencia de codificación según características de imagen o similares, la unidad de codificación máxima puede usarse directamente como la unidad de codificación final o, según sea necesario, la unidad de codificación puede dividirse recursivamente en unidades de codificación de una profundidad más profunda, de manera que puede usarse una unidad de codificación que tenga un tamaño óptimo como la unidad de codificación final. En este caso, el procedimiento de codificación puede incluir un procedimiento como, por ejemplo, predicción, transformada y reconstrucción que se describirán más adelante. Como otro ejemplo, la unidad de procesamiento puede incluir además una unidad de predicción (PU, por sus siglas en inglés) o una unidad de transformada (TU, por sus siglas en inglés). En este caso, cada una de la unidad de predicción y la unidad de transformada puede dividirse o particionarse a partir de la unidad de codificación final descrita anteriormente. La unidad de predicción puede ser una unidad de predicción de muestras, y la unidad de transformada puede ser una unidad para inducir un coeficiente de transformada y/o una unidad para inducir una señal residual a partir del coeficiente de transformada.

La unidad puede usarse indistintamente con el término como, por ejemplo, un bloque o un área en algunos casos. En general, un bloque $M \times N$ puede representar muestras compuestas por M columnas y N filas o un grupo de coeficientes de transformada. La muestra puede representar generalmente un píxel o un valor del píxel, y también puede representar solo el valor de píxel/píxel de un componente de luma, y también representar solo el valor de píxel/píxel de un componente de croma. La muestra puede usarse como el término correspondiente a un píxel o un pel que configura una imagen.

El aparato 200 de codificación puede generar una señal residual (bloque residual, matriz de muestras residuales) restando una señal predicha (bloque predicho, matriz de muestras de predicción) emitida desde el interpredictor 221 o el intrapredictor 222 de la señal de imagen de entrada (bloque original, matriz de muestras originales), y la señal residual generada se transmite al transformador 232. En este caso, como se ilustra, la unidad para restar la señal predicha (bloque predicho, matriz de muestras de predicción) de la señal de imagen de entrada (bloque original, matriz de muestras originales) dentro de un codificador 200 puede denominarse restador 231. El predictor puede llevar a cabo la predicción para un bloque a procesar (en lo sucesivo, denominado bloque actual), y generar un bloque predicho que incluye muestras de predicción del bloque actual. El predictor puede determinar si se aplica intrapredicción o si se aplica interpredicción en unidades del bloque actual o la CU. El predictor puede generar diversa información sobre la predicción como, por ejemplo, información de modo de predicción, para transferir la información generada al codificador 240 de entropía como se describe más adelante en la descripción de cada modo de predicción. La información sobre la predicción puede codificarse por el codificador 240 de entropía para emitirse en forma de flujo de bits.

El intrapredictor 222 puede predecir un bloque actual con referencia a muestras dentro de una imagen actual. Las muestras referenciadas pueden ubicarse vecinas del bloque actual, o también pueden ubicarse alejadas del bloque actual según el modo de predicción. Los modos de predicción en la intrapredicción pueden incluir múltiples modos no direccionales y múltiples modos direccionales. El modo no direccional puede incluir, por ejemplo, un modo de DC o un modo plano. El modo direccional puede incluir, por ejemplo, 33 modos de predicción direccionales o 65 modos de predicción direccionales según el grado fino de la dirección de predicción. Sin embargo, esto es ilustrativo y los modos de predicción direccionales que son más o menos que el número anterior pueden usarse según la configuración. El intrapredictor 222 también puede determinar el modo de predicción aplicado al bloque actual usando el modo de predicción aplicado al bloque vecino.

El interpredictor 221 puede inducir un bloque predicho del bloque actual basándose en un bloque de referencia (matriz de muestras de referencia) especificado por un vector de movimiento en una imagen de referencia. En este momento, para disminuir la cantidad de información de movimiento transmitida en el modo de interpredicción, la información de movimiento puede predecirse en unidades de un bloque, un subbloque o una muestra 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 imagen de referencia. La información de movimiento puede incluir además información de dirección de interpredicción (predicción L0, predicción L1, predicción Bi o similares). En el caso de la interpredicción, el bloque vecino puede incluir un bloque vecino espacial existente dentro de la imagen actual y un bloque vecino temporal existente en la imagen de referencia. La imagen de referencia que incluye el bloque de referencia y la imagen de referencia que incluye el bloque vecino temporal también pueden ser iguales entre sí, y también pueden ser diferentes entre sí. El bloque vecino temporal puede denominarse con un nombre como, por ejemplo, un bloque de referencia co-ubicado, una CU co-ubicada (coCU, por sus siglas en inglés) o similar, y la imagen de referencia que incluye el bloque vecino temporal también puede denominarse imagen co-ubicada (coPic, por sus siglas en inglés). Por ejemplo, el interpredictor 221 puede configurar una lista de candidatos de información de movimiento basándose en los bloques vecinos, y generar información que indica qué candidato se usa para derivar el vector de movimiento y/o el índice de imagen de referencia del bloque actual. La interpredicción puede llevarse a cabo

basándose en diversos modos de predicción y, por ejemplo, en el caso de un modo de salto y un modo de fusión, el interpredictor 221 puede usar la información de movimiento del bloque vecino como la información de movimiento del bloque actual. En el caso del modo de salto, la señal residual puede no transmitirse a diferencia del modo de fusión. Un modo de predicción de vector de movimiento (MVP, por sus siglas en inglés) puede indicar el vector de movimiento del bloque actual usando el vector de movimiento del bloque vecino como un predictor de vector de movimiento, y señalizando una diferencia de vector de movimiento.

El predictor 200 puede generar una señal predicha basándose en diversos métodos de predicción que se describirán más adelante. Por ejemplo, el predictor puede no solo aplicar la intrapredicción o la interpredicción para predecir un bloque, sino también aplicar simultáneamente la intrapredicción y la interpredicción. Esto puede denominarse inter e intrapredicción combinada (CIIP, por sus siglas en inglés). Además, el predictor puede basarse en un modo de predicción de copia intrabloque (IBC, por sus siglas en inglés), o un modo de paleta para llevar a cabo la predicción en un bloque. El modo de predicción IBC o el modo de paleta pueden usarse para la codificación de imagen/vídeo de contenido de un juego o similar como, por ejemplo, la codificación de contenido de pantalla (SCC, por sus siglas en inglés). La IBC lleva a cabo básicamente la predicción en una imagen actual, pero puede llevarse a cabo de manera similar a la interpredicción porque deriva un bloque de referencia en una imagen actual. Es decir, la IBC puede usar al menos una de las técnicas de interpredicción descritas en el presente documento. El modo de paleta puede considerarse como un ejemplo de intracodificación o intrapredicción. Cuando se aplica el modo de paleta, un valor de muestra en una imagen puede señalizarse basándose en información sobre un índice de paleta y una tabla de paleta.

La señal predicha generada a través del predictor (incluidos el interpredictor 221 y/o el intrapredictor 222) puede usarse para generar una señal reconstruida o usarse para generar una señal residual. El transformador 232 puede generar coeficientes de transformada aplicando la 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, por sus siglas en inglés), una transformada de seno discreta (DST, por sus siglas en inglés), una transformada de Karhunen-Loève (KLT, por sus siglas en inglés), una transformada basada en gráficos (GBT, por sus siglas en inglés) o una transformada condicionalmente no lineal (CNT, por sus siglas en inglés). Aquí, cuando la información de relación entre píxeles se ilustra como un gráfico, la GBT significa la transformada obtenida del gráfico. La CNT significa la transformada que se adquiere basándose en una señal predicha generada usando todos los píxeles reconstruidos previamente. Además, el proceso de transformada también puede aplicarse a un bloque de píxeles que tiene el mismo tamaño del cuadrado, y también puede aplicarse al bloque que tiene un tamaño variable en lugar del cuadrado.

El cuantificador 233 puede cuantificar los coeficientes de transformada para transmitir los coeficientes de transformada cuantificados al codificador 240 de entropía, y el codificador 240 de entropía puede codificar la señal cuantificada (información sobre los coeficientes de transformada cuantificados) a la señal cuantificada codificada al flujo de bits. La información sobre los coeficientes de transformada cuantificados puede denominarse información residual. El cuantificador 233 puede reorganizar los coeficientes de transformada cuantificados que tienen una forma de bloque en una forma de vector unidimensional basándose en un orden de escaneo de coeficientes, y también generar la información sobre los coeficientes de transformada cuantificados basándose en los coeficientes de transformada cuantificados de la forma de vector unidimensional. El codificador 240 de entropía puede llevar a cabo diversos métodos de codificación, por ejemplo, una codificación de Golomb exponencial, una codificación de longitud variable adaptativa al contexto (CAVLC, por sus siglas en inglés) y una codificación aritmética binaria adaptativa al contexto (CABAC, por sus siglas en inglés). El codificador 240 de entropía también puede codificar información (p. ej., valores de elementos sintácticos y similares) necesaria para reconstruir video/imagen distinta de los coeficientes de transformada cuantificados juntos o por separado. La información codificada (p. ej., información de vídeo/imagen codificada) puede transmitirse o almacenarse en unidades de la unidad de capa de abstracción de red (NAL, por sus siglas en inglés) en forma de flujo de bits. La información de vídeo/imagen puede incluir además información sobre diversos conjuntos de parámetros como, por ejemplo, un conjunto de parámetros de adaptación (APS, por sus siglas en inglés), un conjunto de parámetros de imagen (PPS, por sus siglas en inglés), un conjunto de parámetros de secuencia (SPS) o un conjunto de parámetros de vídeo (VPS, por sus siglas en inglés). Además, la información de vídeo/imagen puede incluir además información de restricción general. La información señalizada/transmitida y/o los elementos sintácticos que se describirán más adelante en este documento pueden codificarse a través del procedimiento de codificación descrito anteriormente y, por lo tanto, incluirse en el flujo de bits. El flujo de bits puede transmitirse a través de una red, o almacenarse en un medio de almacenamiento digital. En este caso, la red puede incluir una red de radiodifusión y/o una red de comunicación, o similar, y el medio de almacenamiento digital puede incluir diversos medios de almacenamiento como, por ejemplo, USB, SD, CD, DVD, Blue-ray, HDD y SSD. Un transmisor (no se ilustra) para transmitir la señal emitida desde el codificador 240 de entropía y/o un almacenamiento (no se ilustra) para almacenar la señal pueden configurarse como los elementos internos/externos del aparato 200 de codificación, o el transmisor también puede incluirse en el codificador 240 de entropía.

Los coeficientes de transformada cuantificados emitidos desde el cuantificador 233 pueden usarse para generar una señal predicha. Por ejemplo, el descuantificador 234 y el transformador 235 inverso aplican

descuantificación y transformada inversa a los coeficientes de transformada cuantificados, de manera que la señal residual (bloque residual o muestras residuales) puede reconstruirse. El sumador 250 suma la señal residual reconstruida a la señal predicha emitida desde el interpredictor 221 o el intrapredictor 222, de manera que puede generarse la señal reconstruida (imagen reconstruida, bloque reconstruido, matriz de muestras reconstruidas). Como en el caso en el que se aplica el modo de salto, si no hay ningún residuo para el bloque que va a procesarse, el bloque predicho puede usarse como el bloque reconstruido. El sumador 250 puede denominarse reconstructor o generador de bloques reconstruidos. La señal reconstruida generada puede usarse para la intrapredicción del siguiente bloque que va a procesarse dentro de la imagen actual, y tal como se describe más adelante, también usarse para la interpredicción de la siguiente imagen a través de filtrado.

Mientras tanto, también puede aplicarse un mapeo de luma con escalado de croma (LMCS, por sus siglas en inglés) en un proceso de codificación y/o reconstrucción de imágenes.

El filtro 260 puede aplicar filtrado a la señal reconstruida, mejorando por ello las cualidades subjetivas/objetivas de la imagen. Por ejemplo, el filtro 260 puede aplicar diversos métodos de filtrado a la imagen reconstruida para generar una imagen reconstruida modificada, y almacenar la imagen reconstruida modificada en la memoria 270, específicamente, el DPB de la memoria 270. Diversos métodos de filtrado pueden incluir, por ejemplo, un filtrado de desbloqueo, un desplazamiento adaptativo de muestra, un filtro de bucle adaptativo, un filtro bilateral y similares. El filtro 260 puede generar diversa información relacionada con el filtrado para transferir la información generada al codificador 240 de entropía, como se describe más adelante en la descripción de cada método de filtrado. La información relacionada con el filtrado puede ser codificada por el codificador 240 de entropía para ser emitida en forma de flujo de bits.

La imagen reconstruida modificada transmitida a la memoria 270 puede usarse como la imagen de referencia en el interpredictor 221. Si la interpredicción se aplica por el interpredictor, el aparato de codificación puede evitar el desajuste de predicción entre el aparato 200 de codificación y el aparato de decodificación, y también mejorar la eficiencia de codificación.

El DPB de la memoria 270 puede almacenar la imagen reconstruida modificada que se utilizará como imagen de referencia en el interpredictor 221. La memoria 270 puede almacenar información de movimiento del bloque en el que se deriva (o codifica) la información de movimiento dentro de la imagen actual y/o información de movimiento de los bloques dentro de la imagen previamente reconstruida. La información de movimiento almacenada puede transferirse al interpredictor 221 para utilizarse como información de movimiento del bloque vecino espacial o información de movimiento del bloque vecino temporal. La memoria 270 puede almacenar las muestras reconstruidas de los bloques reconstruidos dentro de la imagen actual, y transferir las muestras reconstruidas al intrapredictor 222.

La FIG. 3 es un diagrama para explicar esquemáticamente una configuración de un aparato de decodificación de vídeo/imagen al que es aplicable el presente documento. En lo sucesivo, lo que se denomina aparato de decodificación puede incluir un aparato de decodificación de imágenes y/o un aparato de decodificación de vídeo.

Con referencia a la FIG. 3, el aparato 300 de decodificación puede incluir y configurarse con un decodificador 310 de entropía, un procesador 320 residual, un predictor 330, un sumador 340, un filtro 350 y una memoria 360. El predictor 330 puede incluir un interpredictor 331 y un intrapredictor 332. El procesador 320 residual puede incluir un descuantificador 321 y un transformador 322 inverso. El decodificador 310 de entropía, el procesador 320 residual, el predictor 330, el sumador 340 y el filtro 350, que se han descrito anteriormente, pueden configurarse mediante uno o más componentes de hardware (p. ej., conjuntos de chips o procesadores de decodificador) según una realización. Además, la memoria 360 puede incluir un búfer de imágenes decodificadas (DPB) y puede configurarse mediante un medio de almacenamiento digital. El componente de hardware puede incluir además la memoria 360 como un componente interno/externo.

Cuando se introduce el flujo de bits que incluye la información de vídeo/imagen, el aparato 300 de decodificación puede reconstruir la imagen en respuesta a un proceso en el que la información de vídeo/imagen se procesa en el aparato de codificación ilustrado en la FIG. 2. Por ejemplo, el aparato 300 de decodificación puede derivar las unidades/bloques basándose en información relacionada con la división de bloques adquirida a partir del flujo de bits. El aparato 300 de decodificación puede llevar a cabo la decodificación usando la unidad de procesamiento aplicada al aparato de codificación. Por lo tanto, la unidad de procesamiento para la decodificación puede ser, por ejemplo, una unidad de codificación, y la unidad de codificación puede dividirse según la estructura de árbol cuaternario, la estructura de árbol binario y/o la estructura de árbol ternario de la unidad de árbol de codificación o la unidad de codificación máxima. Una o más unidades de transformada pueden derivarse de la unidad de codificación. Además, la señal de imagen reconstruida decodificada y emitida a través del aparato 300 de decodificación puede reproducirse a través de un aparato de reproducción.

El aparato 300 de decodificación puede recibir la señal emitida desde el aparato de codificación ilustrado en la FIG. 2 en forma de flujo de bits, y la señal recibida puede decodificarse a través del decodificador 310 de entropía. Por ejemplo, el decodificador 310 de entropía puede derivar la información (p. ej., información de

vídeo/imagen) necesaria para la reconstrucción de imágenes analizando sintácticamente el flujo de bits. La información de vídeo/imagen puede incluir además información sobre diversos conjuntos de parámetros como, por ejemplo, un conjunto de parámetros de adaptación (APS), un conjunto de parámetros de imagen (PPS), un conjunto de parámetros de secuencia (SPS) y un conjunto de parámetros de vídeo (VPS). Además, la información de vídeo/imagen puede incluir además información de restricción general. El aparato de decodificación puede decodificar la imagen además basándose en la información sobre el conjunto de parámetros y/o la información de restricción general. La información señalizada/recibida y/o los elementos sintácticos que se describirán más adelante en este documento pueden decodificarse a través del procedimiento de decodificación y adquirirse del flujo de bits. Por ejemplo, el decodificador 310 de entropía puede decodificar información dentro del flujo de bits basándose en un método de codificación como, por ejemplo, una codificación de Golomb exponencial, una CAVLC o una CABAC, y emitir un valor del elemento sintáctico necesario para la reconstrucción de imágenes, y los valores cuantificados del coeficiente de transformada relacionado con residuos. Más específicamente, el método de decodificación de entropía CABAC puede recibir un bin correspondiente a cada elemento sintáctico del flujo de bits, determinar un modelo de contexto usando información de elementos sintácticos a decodificar y decodificar información del bloque vecino y el bloque a decodificar o información del símbolo/bin decodificado en la etapa anterior, y generar un símbolo correspondiente a un valor de cada elemento sintáctico prediciendo la probabilidad de generación del bin según el modelo de contexto determinado para llevar a cabo la decodificación aritmética del bin. En este momento, el método de decodificación de entropía CABAC puede determinar el modelo de contexto y luego actualizar el modelo de contexto usando la información del símbolo/bin decodificado para un modelo de contexto de un siguiente símbolo/bin. La información sobre la predicción entre la información decodificada por el decodificador 310 de entropía puede proveerse al predictor (el interpredictor 332 y el intrapredictor 331), y un valor residual en el que la decodificación de entropía se lleva a cabo por el decodificador 310 de entropía, es decir, los coeficientes de transformada cuantificados y la información de parámetro relacionada pueden introducirse en el procesador 320 residual. El procesador 320 residual puede derivar una señal residual (bloque residual, muestras residuales, matriz de muestras residuales). Además, la información sobre el filtrado entre la información decodificada por el decodificador 310 de entropía puede proveerse al filtro 350. Mientras tanto, un receptor (no se ilustra) para recibir la señal emitida desde el aparato de codificación puede configurarse además como el elemento interno/externo del aparato 300 de decodificación, o el receptor también puede ser un componente del decodificador 310 de entropía. Mientras tanto, el aparato de decodificación según este documento puede denominarse aparato de decodificación de vídeo/imagen, y el aparato de decodificación también puede clasificarse en un decodificador de información (decodificador de información de vídeo/imagen) y un decodificador de muestra (decodificador de muestra de vídeo/imagen). El decodificador de información puede incluir el decodificador 310 de entropía, y el decodificador de muestras puede incluir al menos uno del descuantificador 321, el transformador 322 inverso, el sumador 340, el filtro 350, la memoria 360, el interpredictor 332 y el intrapredictor 331.

El descuantificador 321 puede descuantificar los coeficientes de transformada cuantificados para emitir los coeficientes de transformada. El descuantificador 321 puede reorganizar los coeficientes de transformada cuantificados en una forma de bloque bidimensional. En este caso, la reorganización puede llevarse a cabo basándose en un orden de escaneo de coeficientes llevado a cabo por el aparato de codificación. El descuantificador 321 puede llevar a cabo la descuantificación para los coeficientes de transformada cuantificados usando un parámetro de cuantificación (p. ej., información de tamaño de etapa de cuantificación), y adquirir los coeficientes de transformada.

El transformador 322 inverso transforma inversamente los coeficientes de transformada para adquirir la señal residual (bloque residual, matriz de muestras residuales).

El predictor 330 puede llevar a cabo la predicción del bloque actual y generar un bloque predicho que incluye las muestras de predicción del bloque actual. El predictor puede determinar si se aplica la intrapredicción o si se aplica la interpredicción al bloque actual basándose en la información sobre la predicción emitida desde el decodificador 310 de entropía, y determinar un modo de intrapredicción/interpredicción específico.

El predictor puede generar la señal predicha basándose en diversos métodos de predicción que se describirán más adelante. Por ejemplo, el predictor puede no solo aplicar la intrapredicción o la interpredicción para la predicción de un bloque, sino también aplicar la intrapredicción y la interpredicción al mismo tiempo. Esto puede denominarse inter e intrapredicción combinada (CIIP). Además, el predictor puede basarse en un modo de predicción de copia intrabloque (IBC), o un modo de paleta para llevar a cabo la predicción en un bloque. El modo de predicción de IBC o el modo de paleta pueden usarse para la codificación de imagen/vídeo de contenido de un juego o similar como, por ejemplo, la codificación de contenido de pantalla (SCC). La IBC lleva a cabo básicamente la predicción en una imagen actual, pero puede llevarse a cabo de manera similar a la interpredicción porque deriva un bloque de referencia en una imagen actual. Es decir, la IBC puede usar al menos una de las técnicas de interpredicción descritas en el presente documento. El modo de paleta puede considerarse como un ejemplo de intracodificación o intrapredicción. Cuando se aplica el modo de paleta, la información sobre una tabla de paleta y un índice de paleta puede incluirse en la información de vídeo/imagen y señalizarse.

El intrapredicador 331 puede predecir el bloque actual con referencia a las muestras dentro de la imagen actual. Las muestras referenciadas pueden ubicarse vecinas al bloque actual según el modo de predicción, o también pueden ubicarse lejos del bloque actual. Los modos de predicción en la intrapredicción pueden incluir múltiples modos no direccionales y múltiples modos direccionales. El intrapredicador 331 también puede determinar el modo de predicción aplicado al bloque actual usando el modo de predicción aplicado al bloque vecino.

El interpredicador 332 puede inducir el bloque predicho del bloque actual basándose en el bloque de referencia (matriz de muestras de referencia) especificado por el vector de movimiento en la imagen de referencia. En este momento, para disminuir la cantidad de la información de movimiento transmitida en el modo de interpredicción, la información de movimiento puede predecirse en unidades de un bloque, un subbloque o una muestra 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 imagen de referencia. La información de movimiento puede incluir además información de dirección de interpredicción (predicción L0, predicción L1, predicción Bi o similares). En el caso de la interpredicción, el bloque vecino puede incluir un bloque vecino espacial existente dentro de la imagen actual y un bloque vecino temporal existente en la imagen de referencia. Por ejemplo, el interpredicador 332 puede configurar una lista de candidatos de información de movimiento basándose en los bloques vecinos, y derivar el vector de movimiento y/o el índice de imagen de referencia del bloque actual basándose en la información de selección de candidatos recibida. La interpredicción puede llevarse a cabo basándose en diversos modos de predicción, y la información sobre la predicción puede incluir información que indica el modo de la interpredicción del bloque actual.

El sumador 340 puede agregar la señal residual adquirida a la señal predicha (bloque predicho, matriz de muestras de predicción) emitida desde el predictor (incluidos el interpredicador 332 y/o el intrapredicador 331) para generar la señal reconstruida (imagen reconstruida, bloque reconstruido, matriz de muestras reconstruidas). Como en el caso en el que se aplica el modo de salto, si no hay ningún residuo para el bloque que va a procesarse, el bloque predicho puede usarse como el bloque reconstruido.

El sumador 340 puede denominarse reconstructor o generador de bloques reconstruidos. La señal reconstruida generada puede usarse para la intrapredicción de un siguiente bloque que va a procesarse dentro de la imagen actual, y tal como se describe más adelante, también puede emitirse a través de filtrado o también puede usarse para la interpredicción de una imagen siguiente.

Mientras tanto, también puede aplicarse un mapeo de luma con escalado de croma (LMCS) en el proceso de decodificación de imágenes.

El filtro 350 puede aplicar filtrado a la señal reconstruida, mejorando de este modo las cualidades subjetivas/objetivas de la imagen. Por ejemplo, el filtro 350 puede aplicar diversos métodos de filtrado a la imagen reconstruida para generar una imagen reconstruida modificada, y transmitir la imagen reconstruida modificada a la memoria 360, específicamente, el DPB de la memoria 360. Diversos métodos de filtrado pueden incluir, por ejemplo, un filtrado de desbloqueo, un desplazamiento adaptativo de muestra, un filtro de bucle adaptativo, un filtro bidireccional y similares.

La imagen reconstruida (modificada) almacenada en el DPB de la memoria 360 puede usarse como la imagen de referencia en el interpredicador 332. La memoria 360 puede almacenar información de movimiento del bloque en el que se deriva (decodifica) la información de movimiento dentro de la imagen actual y/o información de movimiento de los bloques dentro de la imagen previamente reconstruida. La información de movimiento almacenada puede transferirse al interpredicador 260 para utilizarse como información de movimiento del bloque vecino espacial o información de movimiento del bloque vecino temporal. La memoria 360 puede almacenar las muestras reconstruidas de los bloques reconstruidos dentro de la imagen actual, y transferir las muestras reconstruidas almacenadas al intrapredicador 331.

En el presente documento, las realizaciones a modo de ejemplo descritas en el filtro 260, el interpredicador 221 y el intrapredicador 222 del aparato 200 de codificación pueden aplicarse igualmente a o para corresponder al filtro 350, al interpredicador 332 y al intrapredicador 331 del aparato 300 de decodificación, respectivamente.

Mientras tanto, como se ha descrito anteriormente, al llevar a cabo la codificación de vídeo, se lleva a cabo la predicción para mejorar la eficiencia de compresión. A través de esto, se puede generar un bloque predicho que incluye muestras de predicción para un bloque actual como un bloque a codificar (es decir, un bloque objetivo de codificación). Aquí, el bloque predicho incluye muestras de predicción en un dominio espacial (o dominio de píxeles). El bloque predicho se deriva de la misma manera en un aparato de codificación y un aparato de decodificación, y el aparato de codificación puede señalar información (información residual) sobre el residuo entre el bloque original y el bloque predicho, en lugar de un valor de muestra original de un bloque original, al aparato de decodificación, aumentando de este modo la eficiencia de codificación de imágenes. El aparato de decodificación puede derivar un bloque residual que incluye muestras residuales en base a la información residual, añadir el bloque residual y el bloque predicho para generar bloques reconstruidos que incluyen muestras reconstruidas y generar una imagen reconstruida que incluye los bloques reconstruidos.

La información residual puede generarse a través de un procedimiento de transformada y cuantificación. Por ejemplo, el aparato de codificación puede derivar un bloque residual entre el bloque original y el bloque predicho, llevar a cabo un procedimiento de transformada en muestras residuales (matriz de muestras residuales) incluidas en el bloque residual para derivar coeficientes de transformada, llevar a cabo un procedimiento de cuantificación en los coeficientes de transformada para derivar coeficientes de transformada cuantificados y señalar información residual relacionada al aparato de decodificación (a través de un flujo de bits). En este caso, la información residual puede incluir información de valor de los coeficientes de transformada cuantificados, información de ubicación, una técnica de transformada, un núcleo de transformada, un parámetro de cuantificación y similares. El aparato de decodificación puede llevar a cabo un procedimiento de descuantificación/transformada inversa en base a la información residual y derivar muestras residuales (o bloques residuales). El aparato de decodificación puede generar una imagen reconstruida basándose en el bloque predicho y el bloque residual. Asimismo, para referencia para la interpredicción de una imagen posteriormente, el aparato de codificación también puede descuantificar/transformar inversamente los coeficientes de transformada cuantificados para derivar un bloque residual y generar una imagen reconstruida basándose en los mismos.

El siguiente dibujo se ha preparado para explicar un ejemplo detallado del presente documento. Dado que el nombre de un dispositivo detallado o un término o nombre detallado (p. ej., nombre de sintaxis) descrito en el dibujo se presenta a modo de ejemplo, las características técnicas del presente documento no se limitan al nombre detallado usado en el dibujo.

La FIG. 4 ilustra un ejemplo de un método de codificación de vídeo/imagen esquemático al que son aplicables realizaciones del presente documento.

El método descrito en la FIG. 4 puede llevarse a cabo por el aparato 200 de codificación de la FIG. 2 como se describió anteriormente. Específicamente, E400 puede llevarse a cabo por el interpredictor 221 o el intrapredictor 222 del aparato 200 de codificación, y E410, E420, E430 y E440 pueden llevarse a cabo por el restador 231, el transformador 232, el cuantificador 233 y el codificador 240 de entropía del aparato 200 de codificación, respectivamente.

Con referencia a la FIG. 4, el aparato de codificación puede derivar muestras de predicción a través de la predicción para el bloque actual (E400). El aparato de codificación puede determinar si llevar a cabo la interpredicción o intrapredicción con respecto al bloque actual, y puede determinar un modo de interpredicción detallado o un modo de intrapredicción detallado basándose en un coste de RD. Según el modo determinado, el aparato de codificación puede derivar las muestras de predicción para el bloque actual.

El aparato de codificación puede derivar muestras residuales a través de la comparación de las muestras de predicción con las muestras originales para el bloque actual (E410).

El aparato de codificación puede derivar coeficientes de transformada a través de un proceso de transformada para muestras residuales (E420), y puede derivar coeficientes de transformada cuantificados cuantificando los coeficientes de transformada derivados (E430).

La cuantificación puede llevarse a cabo basándose en un parámetro de cuantificación. El proceso de transformada y/o el proceso de cuantificación pueden omitirse. En caso de que se omita el proceso de transformada, los coeficientes (cuantificados) (residuales) para las muestras residuales pueden codificarse según una técnica de codificación residual que se describirá más adelante. Para la unidad de los términos, incluso el coeficiente (cuantificado) (residual) puede denominarse coeficiente de transformada (cuantificado).

El aparato de codificación puede codificar información de imagen que incluye información de predicción e información residual, y puede emitir la información de imagen codificada en forma de un flujo de bits (E440). La información de predicción puede ser información relacionada con el proceso de predicción, y puede incluir información (p. ej., en caso de que se aplique interpredicción) sobre la información del modo de predicción y la información de movimiento. La información residual puede incluir información sobre los coeficientes de transformada cuantificados. La información residual puede ser codificada por entropía. Alternativamente, la información residual puede incluir información sobre los coeficientes (cuantificados) (residuales).

El flujo de bits de salida puede transferirse a un aparato de decodificación a través de un medio de almacenamiento o una red.

La FIG. 5 ilustra un ejemplo de un procedimiento esquemático de decodificación de vídeo/imagen al que son aplicables realizaciones del presente documento.

El método descrito en la FIG. 5 puede llevarse a cabo por el aparato 300 de decodificación de la FIG. 3 tal como se describió anteriormente. Específicamente, E500 puede llevarse a cabo por el interpredictor 332 o el intrapredictor 331 del aparato 300 de decodificación. En E500, el decodificador 310 de entropía del aparato 300

de decodificación puede llevar a cabo un proceso de derivación de valores de elementos sintácticos relacionados decodificando información de predicción incluida en el flujo de bits. E510, E520, E530 y E540 pueden llevarse a cabo por el decodificador 310 de entropía, el descuantificador 321, el transformador 322 inverso y el sumador 340 del aparato 300 de decodificación, respectivamente.

5 Con referencia a la FIG. 5, el aparato de decodificación puede llevar a cabo una operación correspondiente a la operación llevada a cabo por el aparato de codificación. El aparato de decodificación puede llevar a cabo la interpredicción o intrapredicción con respecto al bloque actual basándose en la información de predicción recibida, y puede derivar muestras de predicción (E500).

10 El aparato de decodificación puede derivar coeficientes de transformada cuantificados para el bloque actual basándose en la información residual recibida (E510). El aparato de decodificación puede derivar los coeficientes de transformada cuantificados de la información residual a través de la decodificación de entropía.

15 El aparato de decodificación puede derivar los coeficientes de transformada descuantificando los coeficientes de transformada cuantificados (E520). La descuantificación puede llevarse a cabo basándose en el parámetro de cuantificación.

20 El aparato de decodificación puede derivar muestras residuales a través de un proceso de descuantificación para los coeficientes de transformada (E530).

25 El proceso de transformada inversa y/o el proceso de descuantificación pueden omitirse. En caso de que se omita el proceso de transformada inversa, los coeficientes (cuantificados) (residuales) pueden derivarse de la información residual, y las muestras residuales pueden derivarse en base a los coeficientes (cuantificados) (residuales).

30 El aparato de decodificación puede generar muestras reconstruidas para el bloque actual en base a las muestras de predicción y las muestras residuales y, en base a esto, puede generar una imagen reconstruida (E540). A continuación, se puede aplicar además un proceso de filtrado en bucle a la imagen reconstruida como se ha descrito anteriormente.

35 Mientras tanto, como se describió anteriormente, el aparato de codificación puede llevar a cabo la codificación de entropía basándose en diversos métodos de codificación, por ejemplo, Golomb exponencial, codificación de longitud variable adaptativa al contexto (CAVLC) y codificación aritmética binaria adaptativa al contexto (CABAC). Además, el aparato de decodificación puede llevar a cabo una decodificación de entropía basándose en el método de codificación como, por ejemplo, la codificación de Golomb exponencial, CAVLC o CABAC. En lo sucesivo, se describirá un proceso de codificación/decodificación de entropía.

40 La FIG. 6 ilustra esquemáticamente un ejemplo de un método de codificación de entropía al que son aplicables realizaciones del presente documento, y la FIG. 7 ilustra esquemáticamente un codificador de entropía en un aparato de codificación. El codificador de entropía en el aparato de codificación de la FIG. 7 puede aplicarse igualmente o de manera correspondiente incluso al codificador 240 de entropía del aparato 200 de codificación de la FIG. 2 tal como se describió anteriormente.

45 Con referencia a las FIGS. 6 y 7, el aparato de codificación (codificador de entropía) puede llevar a cabo un proceso de codificación de entropía para información de imagen/vídeo. La información de imagen/vídeo puede incluir información relacionada con la partición, información relacionada con la predicción (p. ej., información de clasificación de inter/intrapredicción, información de modo de intrapredicción, información de modo de interpredicción y similares), información residual e información relacionada con el filtrado en bucle, y también puede incluir diversos elementos sintácticos de la misma. La codificación de entropía puede llevarse a cabo en la unidad de un elemento sintáctico. E600 a E610 pueden llevarse a cabo por el codificador 240 de entropía del aparato 200 de codificación de la FIG. 2 como se describió anteriormente.

55 El aparato de codificación puede llevar a cabo una binarización para un elemento sintáctico objetivo (E600). En este caso, la binarización puede basarse en varios métodos de binarización como, por ejemplo, el proceso de binarización de Rice truncado y el proceso de binarización de longitud fija, y el método de binarización para el elemento sintáctico objetivo puede estar predefinido. El proceso de binarización puede llevarse a cabo por el binarizador 242 en el codificador 240 de entropía.

60 El aparato de codificación puede llevar a cabo la codificación de entropía para el elemento sintáctico objetivo (E610). El aparato de codificación puede llevar a cabo la codificación basada en codificación normal (basada en contexto) o basada en codificación de derivación de una cadena de bins del elemento sintáctico objetivo basada en una técnica de codificación de entropía como, por ejemplo, codificación aritmética adaptativa al contexto (CABAC) o codificación de longitud variable adaptativa al contexto (CAVLC), y su salida puede incluirse en el flujo de bits. El proceso de codificación de entropía puede llevarse a cabo por el procesador 243 de codificación de entropía en el codificador 240 de entropía. El flujo de bits puede transferirse al aparato de

65

decodificación a través de un medio de almacenamiento (digital) o una red como se describió anteriormente.

La FIG. 8 ilustra esquemáticamente un ejemplo de un método de decodificación de entropía al que son aplicables realizaciones del presente documento, y la FIG. 9 ilustra esquemáticamente un decodificador de entropía en un aparato de decodificación. El decodificador de entropía en el aparato de decodificación de la FIG. 9 puede aplicarse igualmente o de manera correspondiente incluso al decodificador 310 de entropía del aparato 300 de decodificación de la FIG. 3 tal como se describió anteriormente.

Con referencia a las FIGS. 8 y 9, el aparato de decodificación (decodificador de entropía) puede decodificar información de imagen/vídeo codificada. La información de imagen/vídeo puede incluir información relacionada con la partición, información relacionada con la predicción (p. ej., información de clasificación de inter/intrapredicción, información de modo de intrapredicción, información de modo de interpredicción y similares), información residual e información relacionada con el filtrado en bucle, y también puede incluir diversos elementos sintácticos de la misma. La codificación de entropía puede llevarse a cabo en la unidad de un elemento sintáctico. E800 a E810 pueden llevarse a cabo por el decodificador 310 de entropía del aparato 300 de decodificación de la FIG. 3 como se describió anteriormente.

El aparato de decodificación puede llevar a cabo una binarización para un elemento sintáctico objetivo (E800). En este caso, la binarización puede basarse en varios métodos de binarización como. Por ejemplo, el proceso de binarización de Rice truncado y el proceso de binarización de longitud fija, y el método de binarización para el elemento sintáctico objetivo puede estar predefinido. El aparato de decodificación puede derivar cadenas de bin habilitadas (candidatos de cadena de bin) para valores habilitados del elemento sintáctico objetivo a través del proceso de binarización. El proceso de binarización puede llevarse a cabo por el binarizador 312 en el decodificador 310 de entropía.

El aparato de decodificación puede llevar a cabo una decodificación de entropía para el elemento sintáctico objetivo (E810). El aparato de decodificación compara la cadena de bins derivada con cadenas de bins habilitadas para el elemento sintáctico correspondiente mientras decodifica y analiza secuencialmente los bins respectivos para el elemento sintáctico objetivo a partir del (de los) bit(s) de entrada en el flujo de bits. Si la cadena de bins derivada es igual a una de las cadenas de bins habilitadas, un valor correspondiente a la cadena de bins correspondiente puede derivarse como un valor del elemento sintáctico correspondiente. Si no, el proceso descrito anteriormente puede llevarse a cabo de nuevo después de analizar adicionalmente el siguiente bit en el flujo de bits. A través de dicho proceso, la información correspondiente puede señalizarse usando un bit de longitud variable incluso sin usar un bit inicial o un bit final para información específica (elemento sintáctico específico) en el flujo de bits. A través de esto, se puede asignar un número relativamente menor de bits con respecto a un valor bajo y, por lo tanto, se puede mejorar la eficiencia de codificación global.

El aparato de decodificación puede llevar a cabo una decodificación basada en contexto o basada en derivación de los respectivos bins en la cadena de bins a partir del flujo de bits basándose en la técnica de codificación de entropía como, por ejemplo, CABAC o CAVLC. En este caso, el flujo de bits puede incluir diversos tipos de información para la decodificación de imágenes/vídeo como se ha descrito anteriormente. El flujo de bits puede transferirse al aparato de decodificación a través de un medio de almacenamiento (digital) o una red como se describió anteriormente.

En el presente documento, para representar la señalización de información desde el aparato de codificación al aparato de decodificación, puede usarse una tabla (tabla sintáctica) que incluye elementos sintácticos. El orden de los elementos sintácticos en la tabla sintáctica usada en el presente documento puede representar un orden de análisis sintáctico de los elementos sintácticos del flujo de bits. El aparato de codificación puede configurar y codificar la tabla sintáctica de modo que los elementos sintácticos puedan analizarse por el aparato de decodificación en el orden de análisis, y el aparato de decodificación puede obtener valores de los elementos sintácticos analizando y decodificando los elementos sintácticos en la tabla sintáctica correspondiente en el orden de análisis a partir del flujo de bits.

Mientras tanto, como se ha descrito anteriormente, las muestras residuales pueden derivarse como coeficientes de transformada cuantificados a través de los procesos de transformada y cuantificación. Los coeficientes de transformada cuantificados pueden denominarse coeficientes de transformada. En este caso, los coeficientes de transformada en el bloque pueden señalizarse en forma de información residual. La información residual puede incluir una sintaxis de codificación residual. Es decir, el aparato de codificación puede configurar la sintaxis de codificación residual con la información residual, y puede codificar la sintaxis de codificación residual configurada para emitir la sintaxis de codificación residual codificada en forma de un flujo de bits. El aparato de decodificación puede derivar coeficientes de transformada residuales (cuantificados) decodificando la sintaxis de codificación residual del flujo de bits. Como se describe a continuación, la sintaxis de codificación residual puede incluir elementos sintácticos que representan si la transformada se ha aplicado al bloque correspondiente, donde es la posición del último coeficiente de transformada efectivo en el bloque, si un coeficiente de transformada efectivo está presente en un subbloque y lo que es el tamaño/signo del coeficiente de transformada efectivo.

Por ejemplo, los coeficientes de transformada (cuantificados) pueden codificarse y/o decodificarse basándose en los elementos sintácticos como, por ejemplo, last_sig_coeff_x_prefix, last_sig_coeff_y_prefix, last_sig_coeff_x_suffix, last_sig_coeff_y_suffix, coded_sub_block_flag, sig_coeff_flag, par_level_flag, abs_level_gtX_flag, abs_remainder, coeff_sign_flag, dec_abs_level incluidos en la información residual. Esto puede denominarse codificación residual (datos) o codificación de coeficientes (transformada). En este caso, el proceso de transformada/cuantificación puede omitirse. En este caso, los valores de las muestras residuales pueden codificarse y señalizarse según un método determinado. Los elementos sintácticos relacionados con la codificación/decodificación de datos residuales pueden representarse como en la Tabla 1 a continuación.

5

10

Tabla 1

residual_coding(x0, y0, log2TbWidth, log2TbHeight, cldx) {	Descriptor
si((tu_mts_idx[x0][y0] > 0 (cu_sbt_flag && log2TbWidth < 6 && log2TbHeight < 6)) && cldx == 0 && log2TbWidth > 4)	
log2ZoTbWidth = 4	
de lo contrario	
log2ZoTbWidth = Min(log2TbWidth, 5)	
MaxCbcs = 2 * (1 << log2TbWidth) * (1 << log2TbHeight)	
si(tu_mts_idx[x0][y0] > 0 (cu_sbt_flag && log2TbWidth < 6 && log2TbHeight < 6)) && cldx == 0 && log2TbHeight > 4)	
log2ZoTbHeight = 4	
de lo contrario	
log2ZoTbHeight = Min(log2TbHeight, 5)	
si(log2TbWidth > 0)	
last_sig_coeff_x_prefix	ae(v)
si(log2TbHeight > 0)	
last_sig_coeff_y_prefix	ae(v)
si(last_sig_coeff_x_prefix > 3)	
last_sig_coeff_x_suffix	ae(v)
si(last_sig_coeff_y_prefix > 3)	
last_sig_coeff_y_suffix	ae(v)
log2TbWidth = log2ZoTbWidth	
log2TbHeight = log2ZoTbHeight	
remBinsPass1 = ((1 << (log2TbWidth + log2TbHeight)) * 7) >> 2	
log2SbW = (Min(log2TbWidth, log2TbHeight) < 2 ? 1 : 2)	
log2SbH = log2SbW	
si(log2TbWidth + log2TbHeight > 3) {	
si(log2TbWidth < 2) {	
log2SbW = log2TbWidth	
log2SbH = 4 - log2SbW	
} de lo contrario si (log2TbHeight < 2) {	
log2SbH = log2TbHeight	
log2SbW = 4 - log2SbH	
}	
}	
numSbCoeff = 1 << (log2SbW + log2SbH)	

ES 3 025 935 T3

	Descriptor
residual_coding(x0, y0, log2TbWidth, log2TbHeight, cldx) {	
lastScanPos = numSbCoeff	
lastSubBlock = (1 << (log2TbWidth + log2TbHeight - (log2SbW + log2SbH)) - 1	
hacer {	
si(lastScanPos == 0) {	
lastScanPos = numSbCoeff	
lastSubBlock--	
}	
lastScanPos--	
xS = DiagScanOrder[log2TbWidth - log2SbW] [log2TbHeight - log2SbH] [lastSubBlock] [0]	
yS = DiagScanOrder[log2TbWidth - log2SbW] [log2TbHeight - log2SbH] [lastSubBlock] [1]	
xC = (xS << log2SbW) + DiagScanOrder [log2SbW] [log2SbH] [lastScanPos] [0]	
yC = (yS << log2SbH) + DiagScanOrder [log2SbW] [log2SbH] [lastScanPos] [1]	
} mientras que((xC != LastSignificantCoeffX) (yC != LastSignificantCoeffY))	
si(lastSubBlock == 0 && log2TbWidth >= 2 && log2TbHeight >= 2 && !transform_skip_flag [x0] [y0] && lastScanPos > 0)	
LfnstDcOnly = 0	
si((lastSubBlock > 0 && log2TbWidth >= 2 && log2Tb >= 2) (lastScanPos > 7 && (log2TbWidth == 2 log2TbWidth == 3) && log2TbWidth == log2TbHeight))	
LfnstZeroOutSigCoeffFlag = 0	
QState = 0	
dado que(i = lastSubBlock; i >= 0; i--) {	
startQStateSb = QState	
xS = DiagScanOrder [log2TbWidth - log2SbW] [log2TbHeight - log2SbH] [i] [0]	
yS = DiagScanOrder [log2TbWidth - log2SbW] [log2TbHeight - log2SbH] [i] [1]	
inferSbDcSigCoeffFlag = 0	
si((i < lastSubBlock) && (i > 0)) {	
coded_sub_block_flag[xS][yS]	ae(v)
inferSbDcSigCoeffFlag = 1	
}	
firstSigScanPosSb = numSbCoeff	
lastSigScanPosSb = -1	
firstPosMode0 = (i == lastSubBlock ? lastScanPos : numSbCoeff - 1)	
firstPosMode1 = -1	
dado que(n = firstPosMode0; n >= 0 && remBinsPass1 >= 4; n--) {	
xC = (xS << log2SbW) + DiagScanOrder [log2SbW][log2SbH][n][0]	
yC = (yS << log2SbH) + DiagScanOrder [log2SbW] [log2SbH] [n] [1]	
si(coded_sub_block_flag[xS][yS] && (n > 0 !inferSbDcSigCoeffFlag) && (xC != LastSignificantCoeffX yC != LastSignificantCoeffY)) {	
sig_coeff_flag[xC][yC]	ae(v)
remBinsPass1--	
si(sig_coeff_flag [xC][yC])	

ES 3 025 935 T3

	Descriptor
residual_coding(x0, y0, log2TbWidth, log2TbHeight, cldx) {	
inferSbDcSigCoeffFlag = 0	
}	
si(sig_coeff_flag[xC][yC]) {	
abs_level_gtx_flag[n][0]	ae(v)
remBinsPass1--	
si(abs_level_gtx_flag [n][0]) {	
par_level_flag [n]	ae(v)
remBinsPass1--	
abs_level_gtx_flag[n][1]	ae(v)
remBinsPass1--	
}	
si(lastSigScanPosSb == -1)	
lastSigScanPosSb = n	
firstSigScanPosSb = n	
}	
AbsLevelPass1[xC][yC]= sig_coeff_flag[xC][yC] + par_level_flag[n] +	
abs_level_gtx_flag[n][0] + 2 * abs_level_gtx_flag[n][1]	
si(dep_quant_enabled_flag)	
QState = QStateTransTable [QState] [AbsLevelPass1 [xC] [yC] & 1]	
si(remBinsPass1 < 4)	
firstPosMode1 = n - 1	
}	
dado que(n = numSbCoeff - 1; n >= firstPosMode1; n --) {	
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][n][0]	
yC = (yS << log2SbH) + DiagScanOrder[log2SbW] [log2SbH][n][1]	
si(abs_level_gtx_flag[n][1])	
abs_remainder[n]	ae(v)
AbsLevel[xC][yC]= AbsLevelPass1[xC][yC][+2*abs_remainder[n]	
}	
dado que(n = firstPosMode1; n >= 0; n--) {	
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][n][0]	
yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][n][1]	
dec_abs_level[n]	ae(v)
si(AbsLevel[xC][yC] > 0)	
firstSigScanPosSb = n	
si(dep_quant_enabled_flag)	
QState = QStateTransTable [QState][AbsLevel[xC][yC] & 1]	
}	
si(dep_quant_enabled_flag !sign_data_hiding_enabled_flag)	
signHidden = 0	
de lo contrario	

	Descriptor
residual_coding(x0, y0, log2TbWidth, log2TbHeight, cldx) {	
signHidden = (lastSigScanPosSb - firstSigScanPosSb > 3 ? 1: 0)	
dado que(n = numSbCoeff - 1; n >= 0; n--) {	
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][n][0]	
yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][n][1]	
si((AbsLevel[xC][yC] > 0) && (!signHidden (n != firstSigScanPosSb)))	
coeff_sign_flag[n]	ae(v)
}	
si(dep_quant_enabled_flag) {	
QState = startQStateSb	
dado que(n = numSbCoeff - 1; n >= 0; n--) {	
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][n][0]	
yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][n][1]	
si(AbsLevel[xC][yC] > 0)	
TransCoeffLevel[x0][y0][cldx][xC][yC] = (2 * AbsLevel[xC][yC] - (QState > 1 ? 1:0)) * (1-2 * coeff_sign_flag[n])	
QState = QStateTransTable [QState][par_level_flag[n]]	
} de lo contrario {}	
sumAbsLevel = 0	
dado que(n = numSbCoeff - 1; n >= 0; n--) {	
xC = (xS << log2SbW) + DiagScanOrder[log2SbW][log2SbH][n][0]	
yC = (yS << log2SbH) + DiagScanOrder[log2SbW][log2SbH][n][1]	
si(AbsLevel[xC][yC] > 0) {	
TransCoeffLevel[x0][y0][cldx][xC][yC] = AbsLevel[xC][yC] * (1 - 2 * coeff_sign_flag[n])	
si(signHidden) {	
sumsAbsLevel += AbsLevel[xC][yC]	
si((n == firstSigScanPosSb) && (sumAbsLevel % 2) == 1))	
TransCoeffLevel[x0][y0][cldx][xC][yC] = -TransCoeffLevel[x0][y0][cldx][xC][yC]	
}	
}	
}	
}	
}	

Con referencia a la Tabla 1, last_sig_coeff_x_prefix, last_sig_coeff_y_prefix, last_sig_coeff_x_suffix, y last_sig_coeff_y_suffix son elementos sintácticos para codificar (x, y) información de posición del último coeficiente que no es 0 en un bloque asociado. El bloque asociado puede ser un bloque de codificación (CB) o un bloque de transformada (TB). En relación con la transformada (y cuantificación) y el proceso de codificación residual, el CB y el TB pueden usarse indistintamente. Por ejemplo, se pueden derivar muestras residuales para el CB, y los coeficientes de transformada (cuantificados) se pueden derivar a través de la transformada y cuantificación con respecto a las muestras residuales como se ha descrito anteriormente, y la información (o elementos sintácticos) que representa eficientemente (la posición, el tamaño y el signo de) los coeficientes de transformada (cuantificados) se puede generar y señalar a través del proceso de codificación residual. Los coeficientes de transformada cuantificados pueden denominarse simplemente coeficientes de transformada. En general, si el CB no es mayor que el TB máximo, el tamaño del CB puede ser igual al tamaño del TB y, en

este caso, el bloque objetivo que se transforma (y cuantifica) y codifica mediante el residual puede denominarse CB o TB. Además, si el CB es mayor que el TB máximo, el bloque objetivo que se transforma (y cuantifica) y se codifica mediante el residual puede denominarse TB. En lo sucesivo, aunque se explica que los elementos sintácticos relacionados con la codificación residual se señalizan en la unidad de un bloque de transformada (TB), esto es a modo de ejemplo, y el TB puede usarse indistintamente con el bloque de codificación (CB) como se describió anteriormente.

Mientras tanto, se pueden aplicar diferentes métodos de codificación residual dependiendo de si se aplica un salto de transformada para la codificación residual. Como una realización, si se aplica el salto de transformada puede representarse usando el elemento sintáctico de indicador de salto de transformada, y la codificación residual puede ramificarse según el valor del elemento sintáctico transform_skip_flag del indicador de salto de transformada. Es decir, se pueden usar diferentes elementos sintácticos para la codificación residual basándose en el valor del indicador de salto de transformada (basándose en si se aplica el salto de transformada). La codificación residual que se usa en caso de que no se aplique el salto de transformada (es decir, en caso de que se aplique la transformada) puede denominarse codificación residual regular (RRC, por sus siglas en inglés), y la codificación residual en caso de que se aplique el salto de transformada (es decir, en caso de que no se aplique la transformada) puede denominarse codificación residual de salto de transformada (TSRC, por sus siglas en inglés).

La Tabla 2 a continuación representa un proceso en el que la codificación residual se ramifica basándose en el elemento sintáctico del indicador de salto de transformada.

Tabla 2

transform_unit(x0, y0, tbWidth, tbHeight, treeType, subTuIndex, chType) {	Descriptor
transform_skip_flag[x0][y0][0]	ae(v)
si(!transform_skip_flag[x0][y0][0] slice_ts_residual_coding_disabled_flag)	
residual_coding(x0, y0, Log2(tbWidth), Log2(tbHeight), 0)	
de lo contrario	
residual_ts_coding(x0, y0, Log2(tbWidth), Log2(tbHeight), 0)	
}	
si(tu_cbf_cb[xC][yC] && treeType != DUAL_TREE_LUMA) {	
si(sps_transform_skip_enabled_flag && !BdpcmFlag[x0][y0][1] && wC <= MaxTsSize && hC <= MaxTsSize && !cu_sbt_flag)	
transform_skip_flag[xC][yC][1]	ae(v)
si(!transform_skip_flag[xC][yC][1] slice_ts_residual_coding_disabled_flag)	
residual_coding(xC, yC, Log2(wC), Log2(hC), 1)	
de lo contrario	
residual_ts_coding(xC, yC, Log2(wC), Log2(hC), 1)	
}	
si(tu_cbf_cr[xC][yC] && treeType != DUAL_TREE_LUMA && !(tu_cbf_cb[xC][yC] && tu_joint_cbr_residual_flag[xC][yC])) {	
si(sps_transform_skip_enabled_flag && !BdpcmFlag[x0][y0][2] && wC <= MaxTsSize && hC <= MaxTsSize && !cu_sbt_flag)	
transform_skip_flag[xC][yC][2]	ae(v)
si(!transform_skip_flag[xC][yC][2] slice_ts_residual_coding_disabled_flag)	
residual_coding(xC, yC, Log2(wC), Log2(hC), 2)	
de lo contrario	
residual_ts_coding(xC, yC, Log2(wC), Log2(hC), 2)	
}	
}	

Con referencia a la Tabla 2 anterior, en caso de que no se aplique el salto de transformada (p. ej., en caso de que el valor de transform_skip_flag sea 0), se lleva a cabo la codificación residual regular, y esto puede llevarse a cabo basándose en los elementos sintácticos descritos en la Tabla 1 descrita anteriormente. Además, en caso de que se aplique el salto de transformada (p. ej., en caso de que el valor de transform_skip_flag sea 1), se lleva a cabo la codificación residual de salto de transformada, y esto puede llevarse a cabo basándose en los elementos sintácticos descritos en la Tabla 3 a continuación.

La Tabla 3 a continuación representa los elementos sintácticos para la codificación residual de salto de transformada.

Tabla 3

	Descriptor
residual_ts_coding(x0, y0, log2TbWidth, log2TbHeight, cldx) {	
log2SbSize = (Min(log2TbWidth, log2TbHeight) < 2 ? 1 : 2)	
numSbCoeff = 1 << (log2SbSize << 1)	
lastSubBlock = (1 << (log2TbWidth + log2TbHeight - 2 * log2SbSize)) - 1	
InferSbCf = 1	
MaxCcbcs = 2 * (1 << log2TbWidth) * (1 << log2TbHeight)	
dado que(i = 0; i <= lastSubBlock; i++) {	
xS = DiagScanOrder[log2TbWidth-log2SbSize][log2TbHeight-log2SbSize][i][0]	
yS = DiagScanOrder[log2TbWidth-log2SbSize][log2TbHeight-log2SbSize][i][1]	
si((i != lastSubBlock !inferSbCbf) {	
coded_sub_block_flag[xS][yS]	ae(v)
}	
si(coded_sub_block_flag[xS][yS] && i < lastSubBlock)	
inferSbCf = 0	
/* First scan pass */	
inferSbSigCoeffFlag = 1	
dado que(n = 0; n <= numSbCoeff - 1; n++) {	
xC = (xS << log2SbSize) + DiagScanOrder [log2SbSize][log2SbSize][n][0]	
yC = (yS << log2SbSize) + DiagScanOrder [log2SbSize][log2SbSize][n][1]	
si(coded_sub_block_flag[xS][yS] && (n != numSbCoeff - 1 !inferSbSigCoeffFlag)) {	
sig_coeff_flag[xC][yC]	ae(v)
MaxCcbcs - -	
si(sig_coeff_flag[xC][yC])	
inferSbCoeffFlag = 0	
}	
CoeffSignLevel[xC][yC] = 0	
si(sig_coeff_flag[xC][yC] {	
coeff_sign_flag[n]	ae(v)
MaxCcbcs - -	
CoeffSignLevel[xC][yC] = (coeff_sign_flag[n] > 0 ? -1 : 1)	
abs_level_gtx_flag[n][0]	ae(v)
MaxCcbcs - -	
si(abs_level_gtx_flag[n][0]) {	
par_level_flag[n]	ae(v)

ES 3 025 935 T3

	Descriptor
residual_ts_coding(x0, y0, log2TbWidth, log2TbHeight, cldx) {	
MaxCcbs- -	
}	
}	
AbsLevelPassX[xC][yC] = sig_coeff_flag[xC][yC] + [par_level_flag[n] + abs_level_gtx_flag[n]][0]	
}	
/* Mayor que pase de escaneo X (numGtXFlags=5) */	
dado que(n = 0; n <= numSbCoeff - 1; n++) {	
xC = (xS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][0]	
yC = (yS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][1]	
dado que(j = 1; j < 5; j++) {	
si(abs_level_gtx_flag[n][j] - 1)	
abs_level_gtx_flag[n][j]	ae(v)
MaxCcbs- -	
AbsLevelPassX[xC][yC] += 2 * abs_level_gtx_flag[n][j]	
}	
}	
/* resto de pase de escaneo */	
dado que(n = 0; n <= numSbCoeff - 1; n++) {	
xC = (xS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][0]	
yC = (yS << log2SbSize) + DiagScanOrder[log2SbSize][log2SbSize][n][1]	
si(abs_level_gtx_flag[n][4])	
abs_remainder[n]	ae(v)
si(intra_bdpcm_flag == 0) {	
absRightCoeff = abs(TransCoeffLevel[x0][y0][cldx][xC-1][yC])	
absBelowCoeff = abs(TransCoeffLevel[x0][y0][cldx][xC][yC-1])	
predCoeff = Max(absRightCoeff, absBelowCoeff)	
si(AbsLevelPassX[xC][yC] + abs_remainder[n] == 1 && predCoeff > 0)	
TransCoeffLevel[x0][y0][cldx][xC][yC] = (1-2 * coeff_sign_flag[n]) * predCoeff	
de lo contrario si(AbsLevelPassX[xC][yC] + abs_remainder[n] <= predCoeff)	
TransCoeffLevel[x0][y0][cldx][xC][yC] = (1 - 2 * coeff_sign_flag[n]) * (AbsLevelPassX[xC][yC] + abs_remainder[n] - 1)	
de lo contrario	
TransCoeffLevel[x0][y0][cldx][xC][yC] = (1-2 * coeff_sign_flag[n]) * (AbsLevelPassX[xC][yC] + abs_remainder[n])	
} de lo contrario	
TransCoeffLevel[x0][y0][cldx][xC][yC] = (1-2 * coeff_sign_flag[n]) * (AbsLevelPassX[xC][yC] + abs_remainder[n])	
}	
}	
}	
}	

Por ejemplo, el indicador de salto de transformada que indica si se lleva a cabo el salto de transformada del bloque de transformada puede analizarse, y puede determinarse si el indicador de salto de transformada es 1.

En caso de que el valor del indicador de salto de transformada sea 1, como se muestra en la Tabla 3, los elementos sintácticos sig_coeff_flag, coeff_sign_flag, abs_level_gtx_flag, par_level_flag y/o abs_remainder para el coeficiente residual del bloque de transformada pueden analizarse, y el coeficiente residual puede derivarse basándose en los elementos sintácticos. En este caso, los elementos sintácticos pueden analizarse secuencialmente, y el orden de análisis puede cambiarse. En este caso, abs_level_gtx_flag puede representar abs_level_gt1_flag, abs_level_gt3_flag, abs_level_gt5_flag, abs_level_gt7_flag y/o abs_level_gt9_flag. Por ejemplo, , abs_level_gtx_flag[n][j] puede ser un indicador que representa si un valor absoluto del nivel de coeficiente de transformada (o valor obtenido desplazando el nivel de coeficiente de transformada en 1 a la derecha) es mayor que $(j < < 1) + 1$ en una posición de escaneo n. El $(j < < 1) + 1$ puede ser sustituido por un valor umbral específico como, por ejemplo, un primer valor umbral o un segundo valor umbral en algunos casos.

Además, en caso de que el valor del indicador de salto de transformada sea 0, como se muestra en la Tabla 1, los elementos sintácticos sig_coeff_flag, abs_level_gtx_flag, par_level_flag, abs_remainder, dec_abs_level y coeff_sign_flag para el coeficiente residual del bloque de transformada pueden analizarse, y el coeficiente residual puede derivarse basándose en los elementos sintácticos. En este caso, los elementos sintácticos pueden analizarse secuencialmente, y el orden de análisis puede cambiarse. Aquí, abs_level_gtx_flag puede representar abs_level_gt1_flag y/o abs_level_gt3_flag.

Mientras tanto, como se ha descrito anteriormente, el aparato de codificación puede derivar un bloque residual (muestras residuales) basándose en un bloque predicho (muestras de predicción) a través de predicciones intra/inter/IBC/paleta, y puede derivar coeficientes de transformada cuantificados aplicando transformada y cuantificación con respecto a las muestras residuales derivadas. La información (información residual) sobre los coeficientes de transformada cuantificados puede incluirse en la sintaxis de codificación residual, y puede emitirse en forma de un flujo de bits después de codificarse. El aparato de decodificación puede obtener información (información residual) sobre los coeficientes de transformada cuantificados del flujo de bits, y puede derivar los coeficientes de transformada cuantificados decodificando la información obtenida. El aparato de decodificación puede derivar las muestras residuales a través de descuantificación/transformada inversa en base a los coeficientes de transformada cuantificados. Como se ha descrito anteriormente, al menos una de la cuantificación/descuantificación y/o la transformada/transformada inversa puede omitirse. En caso de que se omita la transformada/transformada inversa, el coeficiente de transformada puede denominarse coeficiente o coeficiente residual, o para la unidad de expresión, todavía puede denominarse coeficiente de transformada. Si omitir la transformada/transformada inversa puede señalizarse basándose en transform_skip_flag. Por ejemplo, si el valor de transform_skip_flag es 1, puede representar que se omite la transformada/transformada inversa, y esto puede denominarse modo de salto de transformada.

En general, en la codificación de vídeo/imagen, la tasa de cuantificación puede cambiarse, y la tasa de compresión puede ajustarse usando la tasa de cuantificación cambiada. Desde el punto de vista de la implementación, se puede usar un parámetro de cuantificación (QP, por sus siglas en inglés) en lugar del uso directo de la tasa de cuantificación teniendo en cuenta la complejidad. Por ejemplo, se puede usar un parámetro de cuantificación de un valor entero en el intervalo de 0 a 63, y cada valor del parámetro de cuantificación puede corresponder a la tasa de cuantificación real. Además, por ejemplo, un parámetro de cuantificación QP_Y para un componente de luma (muestra de luma) y un parámetro de cuantificación QP_C para un componente de croma (muestra de croma) pueden configurarse de manera diferente.

En un proceso de cuantificación, el coeficiente de transformada cuantificado C' puede obtenerse dividiendo un coeficiente de transformada de entrada C por la tasa de cuantificación Q_{etapa}. En este caso, en consideración de la complejidad computacional, la tasa de cuantificación se realiza en una forma de entero multiplicando la tasa de cuantificación por una escala, y se puede llevar a cabo una operación de desplazamiento tanto como el valor correspondiente al valor de escala. Se puede derivar una escala de cuantificación basándose en un producto de la tasa de cuantificación y el valor de escala. Es decir, la escala de cuantificación puede derivarse según el QP. Por ejemplo, la escala de cuantificación puede aplicarse al coeficiente de transformada C y, en base a esto, puede derivarse el coeficiente de transformada cuantificado C'.

El proceso de descuantificación es inverso al proceso de cuantificación, y el coeficiente de transformada cuantificado C' puede multiplicarse por la tasa de cuantificación Q_{etapa} y, basándose en esto, puede obtenerse un coeficiente de transformada reconstruido C". En este caso, se puede obtener una escala de nivel según el parámetro de cuantificación, la escala de nivel se puede aplicar al coeficiente de transformada cuantificado C' y, en base a esto, se puede derivar el coeficiente de transformada reconstruido C". El coeficiente de transformada reconstruido C" es algo diferente del coeficiente de transformada inicial C debido a una pérdida en el proceso de transformada y/o cuantificación. Por consiguiente, incluso el aparato de codificación lleva a cabo la descuantificación de la misma manera que en el aparato de decodificación.

Además, al llevar a cabo la predicción, puede basarse en la codificación de paleta. La codificación de paleta es una tecnología útil para representar bloques que incluyen un pequeño número de valores de color únicos. En lugar de aplicar la predicción y la transformada al bloque, en un modo de paleta, se señala un índice para representar el valor de cada muestra. El modo de paleta es útil para guardar un espacio de búfer de memoria

de vídeo. El bloque puede codificarse usando el modo de paleta (p. ej., MODE_PLT). Para decodificar el bloque codificado como se ha descrito anteriormente, el decodificador debe decodificar una entrada de paleta y un índice. La entrada de paleta puede representarse por una tabla de paleta, y puede codificarse por una herramienta de codificación de tabla de paleta.

5

La codificación de paleta puede denominarse un modo de paleta (intra) o un modo de codificación de paleta (intra). El bloque actual puede reconstruirse según la codificación de paleta o el modo de paleta. La codificación de paleta puede verse como un ejemplo de intracodificación, o puede verse como uno de los métodos de intrapredicción. Sin embargo, de una manera similar al modo de salto descrito anteriormente, puede no señalizarse un valor residual separado para el bloque correspondiente.

10

Por ejemplo, en caso de que se seleccione el modo de paleta, puede señalizarse información sobre una tabla de paleta. La tabla de paleta puede incluir un índice correspondiente a cada píxel. La tabla de paleta puede configurar una tabla de predicción de paleta a partir de valores de píxeles usados en el bloque anterior. Por ejemplo, los valores de píxel usados previamente pueden almacenarse en un búfer específico (predictor de paleta), y la información de predictor de paleta `palette_predictor_run` para configurar la paleta actual puede recibirse del búfer. Es decir, el predictor de paleta puede incluir datos que representan un índice para al menos una parte de un mapa de índices de paleta del bloque actual. En caso de que la entrada de paleta para expresar el bloque actual no sea suficiente como una entrada de predicción de paleta configurada a partir del predictor de paleta, la información de píxel en la entrada de paleta actual puede transmitirse por separado.

15

20

El modo de paleta puede señalizarse a un nivel de CU, y puede usarse generalmente en caso de que la mayoría de los píxeles en la CU puedan representarse como un conjunto de valores de píxel representativos. Es decir, en el modo de paleta, las muestras en la CU pueden expresarse como un conjunto de valores de píxel representativos. Dicho conjunto puede denominarse paleta. En el caso de una muestra que tiene un valor cercano al valor de píxel en la paleta, se puede señalar un índice de paleta `palette_idx_idc` correspondiente al valor de píxel en la paleta o información capaz de indicar el índice (`run_copy_flag`, `copy_above_palette_indices_flag`). En el caso de una muestra que tiene un valor de píxel que excluye la entrada de paleta, la muestra puede indicarse como un símbolo de escape, y un valor de muestra cuantificado puede señalizarse directamente. En el presente documento, el píxel o el valor de píxel pueden denominarse muestra o valor de muestra.

25

30

Para decodificar un bloque codificado en un modo de paleta, el decodificador requiere información de entrada de paleta e información de índice de paleta. En caso de que el índice de paleta corresponda al símbolo de escape, un valor de escape (cuantificado) puede señalizarse como un componente adicional. Además, el codificador debe derivar una paleta adecuada para la CU correspondiente, y transferir la paleta al decodificador.

35

Para una codificación eficiente de la entrada de paleta, se puede mantener un predictor de paleta. El predictor de paleta y el tamaño máximo de la paleta pueden señalizarse a través de SPS. Además, el predictor de paleta y el tamaño máximo de la paleta pueden estar predefinidos. Por ejemplo, el predictor de paleta y el tamaño máximo de la paleta pueden definirse como 31 y 15, respectivamente, dependiendo de si el bloque actual es un único árbol o un árbol dual. En las normas VVC, se puede transmitir `sps_palette_enabled_flag` que representa si el modo de paleta está habilitado. A continuación, se puede transmitir un indicador de `pred_mode_plt_coding` que representa si la unidad de codificación actual está codificada en el modo de paleta. El predictor de paleta puede inicializarse al comienzo de cada ladrillo o cada segmento.

40

45

Con respecto a cada entrada en el predictor de paleta, se puede señalar un indicador de reutilización, y puede representar si la entrada es parte de la paleta actual. El indicador de reutilización puede transmitirse usando codificación de longitud de ejecución de 0. A partir de entonces, el número de nuevas entradas de paleta puede señalizarse usando codificación de Golomb exponencial de 0^{ésimo}-orden. Por último, se puede señalar un valor de componente para la nueva entrada de paleta. Después de codificar la CU actual, el predictor de paleta puede actualizarse usando la paleta actual, y la entrada del predictor de paleta anterior que no se reutiliza en la paleta actual puede añadirse al final del nuevo predictor de paleta (relleno de paleta) hasta que alcance el tamaño máximo permisible.

50

55

Para codificar un mapa de índices de paleta, el índice puede codificarse usando escaneos transversales horizontales y verticales. El orden de escaneo puede señalizarse explícitamente a partir del flujo de bits usando la información de indicador (p. ej., `palette_transpose_flag`).

Mientras tanto, el índice de paleta puede codificarse usando dos tipos de modos de muestra de paleta y, por ejemplo, pueden usarse el modo "INDEX" y el modo "COPY_ABOVE". Tales modos de paleta pueden señalizarse usando un indicador que representa si el modo de paleta es el modo "INDEX" o el modo "COPY_ABOVE". En este caso, el símbolo de escape puede señalizarse en el modo "INDEX", y puede asignarse un índice que tiene el mismo tamaño que el tamaño de paleta actual. Por ejemplo, si se supone que el tamaño de la paleta actual es 10, los índices n.^o 0 a n.^o 9 pueden significar índices de entrada en la paleta, y el índice n.^o 10 puede significar un índice para el símbolo de escape. En caso de que se utilice el escaneo horizontal, el indicador puede señalizarse excluyendo la fila superior, y en caso de que se utilice el escaneo

60

65

vertical, o el modo anterior sea el modo "COPY_ABOVE", el indicador puede señalizarse excluyendo la primera columna. En el modo "COPY_ABOVE", el índice de paleta de la muestra de la fila en la parte superior puede copiarse. En el modo "INDEX", el índice de paleta puede señalizarse explícitamente. Tanto en el modo "INDEX" como en el modo "COPY_ABOVE", puede señalizarse el valor de ejecución que representa el número de muestras siguientes que se codifican usando el mismo modo. En caso de que el símbolo de escape sea una parte de la ejecución en el modo "INDEX" o el modo "COPY_ABOVE", los valores del componente de escape pueden señalizarse con respecto a cada símbolo de escape.

La codificación del índice de paleta puede ser como sigue. En primer lugar, se puede señalar el número de índices para la CU. A continuación, los índices reales para toda la CU pueden señalizarse usando codificación de longitud fija. El número de índices y el índice pueden codificarse en el modo de derivación. A través de esto, los bins de derivación relacionados con el índice pueden agruparse juntos. A continuación, el modo de muestra de paleta copy_above_palette_indices_flag y la ejecución pueden señalizarse de manera intercalada. Por último, los valores de escape de componente correspondientes a las muestras de escape para toda la CU pueden agruparse y pueden codificarse en el modo de derivación.

La Tabla 4 a continuación representa un ejemplo de una estructura sintáctica que incluye elementos sintácticos relacionados con la codificación basada en modo de paleta para la unidad de codificación, y la Tabla 5 a continuación representa la semántica para elementos sintácticos incluidos en la sintaxis de la Tabla 4.

Tabla 4

palette_coding(x0, y0, cbWidth, cbHeight, treeType) {	Descriptor
startComp = (treeType == DUAL_TREE_CHROMA) ? 1: 0	
numComps = (treeType == SINGLE_TREE) ? (sps_chroma_format_idc == 0 ? 1: 3): (treeType == DUAL_TREE_CHROMA) ? 2: 1	
maxNumPaletteEntries = (treeType == SINGLE_TREE) ? 31: 15	
palettePredictionFinished = 0	
NumPredictedPaletteEntries = 0	
dado que(predictorEntryIdx = 0; predictorEntryIdx < PredictorPaletteSize[startComp] && !palettePredictionFinished && NumPredictedPaletteEntries < maxNumPaletteEntries; predictorEntryIdx++) {	
palette_predictor_run	ae(v)
si(palette_predictor_run != 1) {	
si(palette_predictor_run > 1)	
predictorEntryIdx += palette_predictor_run - 1	
PalettePredictorEntryReuseFlags[predictorEntryIdx] = 1	
NumPredictedPaletteEntries++	
} de lo contrario	
palettePredictionFinished = 1	
}	
si(NumPredictedPaletteEntries < maxNumPaletteEntries)	
num_signalled_palette_entries	ae(v)
dado que(cldx = startComp; cldx < (startComp + numComps); cldx++)	
dado que(i = 0; i < num_signalled_palette_entries; i++)	
new_palette_entries[cldx][i]	ae(v)
si(CurrentPaletteSize[startComp] > 0)	
palette_escape_val_present_flag	ae(v)
si(MaxPaletteIndex > 0) {	
ajuste = 0	
palette_transpose_flag	ae(v)

	Descriptor
palette_coding(x0, y0, cbWidth, cbHeight, treeType) {	
}	
si(treeType != DUAL_TREE_CHROMA && palette_escape_val_present_flag)	
si(pps_cu_qp_delta_enabled_flag && !IsCuQpDeltaCoded) {	
cu_qp_delta_abs	ae(v)
si(cu_qp_delta_abs)	
cu_qp_delta_sign_flag	ae(v)
}	
si(treeType != DUAL_TREE_LUMA && palette_escape_val_present_flag)	
si(sh_cu_chroma_qp_offset_enabled_flag && !IsCuChromaQpOffsetCoded) {	
cu_chroma_qp_offset_flag	ae(v)
si(cu_chroma_qp_offset_flag && pps_chroma_qp_offset_list_len_minus1 > 0)	
cu_chroma_qp_offset_idx	ae(v)
}	
PreviousRunPosition = 0	
PreviousRunType = 0	
dado que(subSetId = 0; subSetId <= (cbWidth * cbHeight - 1) / 16; subSetId++) {	
minSubPos = subSetId * 16	
si(minSubPos + 16 > cbWidth * cbHeight)	
maxSubPos = cbWidth * cbHeight	
de lo contrario	
maxSubPos = minSubPos + 16	
RunCopyMap[x0][y0] = 0	
PaletteScanPos = minSubPos	
log2CbWidth = Log2(cbWidth)	
log2CbHeight = Log2(cbHeight)	
mientras que(PaletteScanPos < maxSubPos) {	
xC = x0+TraverseScanOrder[log2CbWidth][log2CbHeight][PaletteScanPos][0]	
yC = y0+TraverseScanOrder[log2CbWidth][log2CbHeight][PaletteScanPos][1]	
si(PaletteScanPos > 0) {	
xcPrev = x0 + TraverseScanOrder[log2CbWidth][log2CbHeight][PaletteScanPos-1][0]	
ycPrev = y0 + TraverseScanOrder[log2CbWidth][log2CbHeight][PaletteScanPos-1][1]	
}	
si(MaxPaletteIndex > 0 && PaletteScanPos > 0) {	
run_copy_flag	ae(v)
RunCopyMap[xC][yC] = run_copy_flag	
}	
CopyAboveIndicesFlag[xC][yC] = 0	
si(MaxPaletteIndex > 0 && !RunCopyMap[xC][yC]) {	
si(((!palette_transpose_flag && yC > y0) (palette_transpose_flag && xC > x0)) && CopyAboveIndicesFlag[xcPrev][ycPrev] == 0 && PaletteScanPos > 0) {	
copy_above_palette_indices_flag	ae(v)

	Descriptor
palette_coding(x0, y0, cbWidth, cbHeight, treeType) {	
CopyAboveIndicesFlag[xC][yC] = copy_above_palette_indices_flag	
}	
PreviousRunType = CopyAboveIndicesFlag[xC][yC]	
PreviousRunPosition = PaletteScanPos	
} de lo contrario si(PaletteScanPos > 0)	
CopyAboveIndicesFlag[xC][yC] = CopyAboveIndicesFlag[xcPrev][ycPrev]	
PaletteScanPos ++	
}	
PaletteScanPos = minSubPos	
mientras que(PaletteScanPos < maxSubPos) {	
xC = x0 +	
TraverseScanOrder[log2CbWidth][log2CbHeight][PaletteScanPos][0]	
yC = y0 +	
TraverseScanOrder[log2CbWidth][log2CbHeight][PaletteScanPos][1]	
si(PaletteScanPos > 0) {	
xcPrev = x0 +	
TraverseScanOrder[log2CbWidth][log2CbHeight][PaletteScanPos-1][0]	
ycPrev = y0 +	
TraverseScanOrder[log2CbWidth][log2CbHeight][PaletteScanPos-1][1]	
}	
si(MaxPaletteIndex > 0 && !RunCopyMap[xC][yC] && CopyAboveIndicesFlag[xC][yC] = 0)	
{	
si(MaxPaletteIndex - ajustar > 0)	
palette_idx_idc	ae(v)
ajuste = 1	
}	
si(!RunCopyMap[xC][yC] && CopyAboveIndicesFlag[xC][yC] == 0)	
CurrPaletteIndex = palette_idx_idc	
si(CopyAboveIndicesFlag[xC][yC] == 0)	
PaletteIndexMap[xC][yC] = CurrPaletteIndex	
de lo contrario si(!palette_transpose_flag)	
PaletteIndexMap[xC][yC] = PaletteIndexMap[xC][yC-1]	
de lo contrario	
PaletteIndexMap[xC][yC] = PaletteIndexMap[xC-1][yC]	
PaletteScanPos ++	
}	
si(palette_escape_val_present_flag) {	
dado que(cldx = startComp; cldx < (startComp + numComps); cldx++) {	
dado que(sPos = minSubPos; sPos < maxSubPos; sPos++) {	
xC = x0 + TraverseScanOrder[log2CbWidth][log2CbHeight][sPos][0]	
yC = y0 + TraverseScanOrder[log2CbWidth][log2CbHeight][sPos][1]	
si(!(treeType == SINGLE_TREE && cldx != 0 && (xC % SubWidthC != 0 yC % SubHeightC != 0))) {	

ES 3 025 935 T3

palette_coding(x0, y0, cbWidth, cbHeight, treeType) {	Descriptor
si(PaletteIndexMap[cldx][xC][yC] == MaxPaletteIndex) {	
palette_escape_val	ae(v)
PaletteEscapeVal[cldx][xC][yC] = palette_escape_val	
}	
}	
}	
}	
}	
}	
}	

Tabla 5

En la siguiente semántica, los índices de matriz $x0$, $y0$ especifican la ubicación ($x0$, $y0$) de la muestra de luma superior izquierda del bloque de codificación considerado con respecto a la muestra de luma superior izquierda de la imagen. Los índices de matriz xC , yC especifican la ubicación (xC , yC) de la muestra con respecto a la muestra de luma superior izquierda de la imagen, cuando `treeType` es igual a `SINGLE_TREE` o `DUAL_TREE_LUMA`; y con respecto a la muestra de croma superior izquierda de la imagen, cuando `treeType` es igual a `DUAL_TREE_CHROMA`. El índice de matriz `startComp` especifica el primer componente de color de la tabla de paleta actual. `startComp` igual a 0 indica el componente Y, `startComp` igual a 1 indica el componente Cb; `startComp` igual a 2 indica el componente Cr. `numComps` especifica el número de componentes de color en la tabla de paleta actual.

La paleta de predictor consiste en entradas de paleta de unidades de codificación previas que se usan para predecir las entradas en la paleta actual.

`PredictorPaletteSize[startComp]` especifica el tamaño de la paleta de predictor para el primer componente de color de la tabla de paleta actual `startComp`.

`PredictorPaletteSize[startComp]` se deriva como se especifica en la subcláusula 8.4.5.3.

`PalettePredictorEntryReuseFlags[i]` igual a 1 especifica que la i -ésima entrada en la paleta de predictor se reutiliza en la paleta actual.

`PalettePredictorEntryReuseFlags[i]` igual a 0 especifica que la i -ésima entrada en la paleta de predictor no es una entrada en la paleta actual.

Todos los elementos de la matriz `PalettePredictorEntryReuseFlags[i]` se inicializan en 0.

`palette_predictor_run` se usa para determinar el número de ceros que preceden a una entrada distinta de cero en la matriz

`PalettePredictorEntryReuseFlags`.

Es un requisito de conformidad del flujo de bits que el valor de `palette_predictor_run` esté en el rango de 0 a

$(\text{PredictorPaletteSize}[\text{startComp}] - \text{predictorEntryIdx})$, inclusive, donde `predictorEntryIdx` corresponde a la posición actual en la matriz

`PalettePredictorEntryReuseFlags`. La variable `NumPredictedPaletteEntries` especifica el número de entradas en la paleta actual que se reutilizan a partir de la paleta de predictor. El valor de `NumPredictedPaletteEntries` estará en el rango de 0 a `maxNumPaletteEntries`, inclusive.

num_signalled_palette_entries especifica el número de entradas en la paleta actual que se señalizan explícitamente para el primer componente de color de la tabla de paleta actual startComp. Cuando num_signalled_palette_entries no está presente, se infiere que es igual a 0.

La variable CurrentPaletteSize[startComp] especifica el tamaño de la paleta actual para el primer componente de color de la tabla de paleta actual startComp y se deriva como sigue:

```
CurrentPaletteSize[startComp] =
NumPredictedPaletteEntries +
num_signalled_palette_entries (176)
```

El valor de CurrentPaletteSize[startComp] estará en el rango de 0 a maxNumPaletteEntries, inclusive.

new_palette_entries[cidx][i] especifica el valor para la i-ésima entrada de paleta señalizada para el componente de color cidx.

La variable LocalDualTreeFlag se deriva de la siguiente manera:

```
LocalDualTreeFlag = (treeType != SINGLE_TREE &&
(sh_slice_type != [ 0 (sh_slice_type = 0) ] &&
sps_qtbtfl_dual_tree_intra_flag == 0)) ? 1:0 (177)
```

La variable PredictorPaletteEntries[cidx][i] especifica el i-ésimo elemento en la paleta de predictor para el componente de color cidx.

La variable CurrentPaletteEntries[cidx][i] especifica el i-ésimo elemento en la paleta actual para el componente de color cidx y se deriva de la siguiente manera:

```
numPredictedPaletteEntries = 0
dado que(i = 0; i < PredictorPaletteSize[startComp]; i++)
si(PalettePredictorEntryReuseFlags[i]) {
dado que(cidx = LocalDualTreeFlag ? 0 : startComp;
cidx < LocalDualTreeFlag ? 3 : (startComp +
numComps); cidx++)
CurrentPaletteEntries[cidx]
[numPredictorPaletteEntries] =
PredictorPaletteEntries[cidx][i]
```

```

numPredictedPaletteEntries++
}
dado que(cidx = startComp, cidx < (startComp + numComps); cidx++)
(178)
    dado que(i = 0; i < num_signalled_palette_entries; i++)
        CurrentPaletteEntries[cidx]
[numPredictedPaletteEntries + i] :=
                                new_palette_entries[cidx][i]

```

palette_escape_val_present_flag igual a 1 especifica que la unidad de codificación actual contiene al menos una muestra codificada de escape. palette_escape_val_present_flag igual a 0 especifica que no hay muestras codificadas de escape en la unidad de codificación actual. Cuando no está presente, se infiere que el valor de palette_escape_val_present_flag es igual a 1.

La variable MaxPaletteIndex especifica el máximo valor posible para un índice de paleta para la unidad de codificación actual. El valor de MaxPaletteIndex se establece igual a CurrentPaletteSize(startComp) - 1 + palette_escape_val_present_flag.

palette_idx_idx es una indicación de un índice a la table de paleta, CurrentPaletteEntries. El valor de palette_idx_idx estará en el rango de 0 a MaxPaletteIndex, inclusive, para el primer índice en el bloque y en el rango de 0 a (MaxPaletteIndex - 1), inclusive, para los índices restantes en el bloque.

Cuando palette_idx_idx no está presente, se infiere que es igual a 0.

palette_transpose_flag igual a 1 especifica que el escaneo transversal vertical se aplica para el escaneo de los índices para muestras en la unidad de codificación actual. palette_transpose_flag igual a 0 especifica que el escaneo transversal horizontal se aplica para el escaneo de los índices para muestras en la unidad de codificación actual. Cuando no está presente, se infiere que el valor de palette_transpose_flag es igual a 0.

La matriz TraverseScanOrder especifica la matriz de orden de escaneo para la codificación de paleta. Si palette_transpose_flag es igual a 0, a TraverseScanOrder se le asigna el orden de escaneo horizontal HorTravScanOrder. De lo contrario, (palette_transpose_flag es igual a 1), a TraverseScanOrder se le asigna el orden de escaneo vertical VerTravScanOrder.

run_copy_flag igual a 1 especifica que el tipo de ejecución de paleta es el mismo que el tipo de ejecución en la posición previamente escaneada y el índice de paleta es el mismo que el índice en la posición previa escaneada si CopyAboveIndicesFlag[xC][yC] es igual a 0.

De lo contrario, `run_copy_flag` igual a 0 especifica que el tipo de ejecución de paleta es diferente del tipo de ejecución en la pasada inmediatamente precedente.

`copy_above_palette_indices_flag` igual a 1 especifica que el índice de paleta es igual al índice de paleta en la misma ubicación en la fila de más arriba si se usa el escaneo transversal horizontal o la misma ubicación en la columna izquierda si se usa el escaneo transversal vertical.

`copy_above_palette_indices_flag` igual a 0 especifica que una indicación del índice de paleta de la muestra se codifica en el flujo de bits o se infiere.

La variable `CopyAboveIndicesFlag[xC][yC]` igual a 1 especifica que el índice de paleta se copia del índice de paleta en la fila de más arriba (escaneo horizontal) o columna izquierda (escaneo vertical).

`CopyAboveIndicesFlag[xC][yC]` igual a 0 especifica que el índice de paleta se codifica explícitamente en el flujo de bits o se infiere.

La variable `PaletteIndicesMap[xC][yC]` especifica un índice de paleta, que es un índice a la matriz representada por `CurrentPaletteEntries`. El valor de `PaletteIndicesMap[xC][yC]` estará en el rango de 0 a `MaxPaletteIndex`, inclusive.

La variable `adjustedRefPaletteIndex` se deriva de la siguiente manera:

```

adjustedRefPaletteIndex = MaxPaletteIndex + 1
si ( PaletteEntries > 0 ) {
    xPrev = xC +
        TransverseScanOrder[log2(CbWidth)][log2(HdDepth)][PaletteScanPos-1][0]
    yPrev = yC +
        TransverseScanOrder[log2(CbWidth)][log2(HdDepth)][PaletteScanPos-1][1]
    si ( CopyAboveIndicesFlag[xPrev][yPrev] == 0 )
        adjustedRefPaletteIndex = PaletteIndicesMap[xPrev][yPrev]    (179)
    de lo contrario {
        si ( palette_transpose_flag )
            adjustedRefPaletteIndex = PaletteIndicesMap[ xC ][ yC - 1 ]
        de lo contrario
            adjustedRefPaletteIndex = PaletteIndicesMap[ xC - 1 ][ yC ]
    }
}

```

Cuando `CopyAboveIndicesFlag[xC][yC]` es igual a 0, la variable `CurPaletteIndex`

se deriva de la siguiente manera:

```

si ( CurPaletteIndex >= adjustedRefPaletteIndex )
    CurPaletteIndex++    (180)

```

palette_escape_val especifica el valor de muestra codificado de escape cuantificado para un componente.

La variable `PaletteEscapeVal[cIdx][xC][yC]` especifica el valor de escape de una muestra para la cual `PaletteIndicesMap[xC][yC]` es igual a `MaxPaletteIndex` y `palette_escape_val_present_flag` es igual a 1. El índice de matriz `cIdx` especifica el componente de color.

Es un requisito de conformidad del flujo de bits que `PaletteEscapeVal[cIdx][xC][yC]` esté en el rango de 0 a $(1 \ll \text{BitDepth}) - 1$, inclusive.

5 Con referencia a la Tabla 4 y la Tabla 5 anteriores, en caso de que se aplique un modo de paleta para el bloque actual (es decir, la unidad de codificación actual), la sintaxis de codificación de paleta (p. ej., `palette_coding()`) como en la Tabla 4 anterior puede analizarse/señalizarse.

10 Por ejemplo, la tabla de paleta puede configurarse en base a la información de entrada de paleta. La información de entrada de paleta puede incluir elementos sintácticos como, por ejemplo, `palette_predictor_run`, `num_signalled_palette_entries`, y `new_palette_entries`.

Además, un mapa de índices de paleta para el bloque actual puede configurarse en base a la información de índices de paleta. La información de índice de paleta puede incluir elementos sintácticos como, por ejemplo, `num_palette_indices_minus1`, `palette_idx_idc`, y `palette_transpose_flag`. En base a la información de índice de

paleta como se describió anteriormente, el mapa de índices de paleta (p. ej., PaletteIndexMap) puede configurarse derivando el índice de paleta (p. ej., PaletteIndexIdc) para las muestras en el bloque actual mientras circula según una dirección de escaneo transversal (dirección vertical o dirección horizontal).

5 Además, en base al mapa de índices de paleta, se puede obtener un valor de muestra para la entrada de paleta en la tabla de paleta, y se pueden generar muestras reconstruidas del bloque actual en base al valor de muestra mapeado en la entrada de paleta.

10 Además, en caso de que esté presente una muestra que tiene un valor de escape (es decir, en caso de que el valor de palette_escape_val_present_flag sea 1) en el bloque actual, el valor de escape para el bloque actual puede derivarse basándose en la información de escape. La información de escape puede incluir elementos sintácticos como, por ejemplo, palette_escape_val_present_flag y palette_escape_val. Por ejemplo, basándose en la información de valor de escape cuantificada (p. ej., palette_escape_val), puede derivarse el valor de escape para la muestra codificada de escape en el bloque actual. Las muestras reconstruidas del bloque actual
15 pueden generarse en base al valor de escape.

Mientras tanto, en el proceso de codificación/decodificación, se puede usar la técnica de modulación codificada por pulsos diferencial de bloque o modulación de código de pulsos delta basada en bloques (BDPCM, por sus siglas en inglés). La BDPCM puede denominarse modulación de código de pulso delta basada en bloques
20 residuales cuantificada (RDPCM, por sus siglas en inglés).

En caso de que el bloque se prediga aplicando la BDPCM, pueden utilizarse muestras reconstruidas con el fin de predecir filas o columnas del bloque línea por línea. En este caso, una muestra de referencia usada puede ser una muestra no filtrada. La dirección de BDPCM puede representar si se ha usado una predicción de
25 dirección vertical o de dirección horizontal. Es decir, en caso de que se aplique la BDPCM, la dirección vertical o la dirección horizontal se puede seleccionar como la dirección de BDPCM, y la predicción se puede llevar a cabo en la dirección de BDPCM. Un error de predicción puede cuantificarse en un dominio espacial, y la muestra puede reconstruirse añadiendo el error de predicción descuantificado a la predicción (es decir, muestra de predicción). El error de predicción puede significar el residuo. La BDPCM de dominio residual cuantificado puede proponerse como una alternativa de la BDPCM, y la dirección o señalización de predicción puede ser
30 igual a la BDPCM aplicada al dominio espacial. Es decir, el residuo puede reconstruirse a través de la descuantificación después de que los propios coeficientes de cuantificación se acumulen como modulación de código de pulso delta (DPCM, por sus siglas en inglés) a través de la BDPCM de dominio residual cuantificado. Por consiguiente, la BDPCM de dominio residual cuantificado puede usarse con el significado de que un extremo de codificación residual aplica la DPCM. El dominio residual cuantificado utilizado a continuación se obtiene cuantificando el residuo derivado en base a la predicción sin ser transformado, y significa un dominio para la muestra residual cuantificada. Por ejemplo, el dominio residual cuantificado puede incluir el residuo cuantificado (o coeficiente residual cuantificado) al que se aplica el salto de transformada, es decir, la transformada se salta con respecto a la muestra residual, pero la cuantificación se aplica a la misma. Además,
40 por ejemplo, el dominio residual cuantificado puede incluir el coeficiente de transformada cuantificado.

Como se ha descrito anteriormente, la BDPCM puede aplicarse al dominio residual cuantificado, el dominio residual cuantificado puede incluir el residuo cuantificado (o coeficiente residual cuantificado) y, en este caso, el salto de transformada puede aplicarse con respecto al residuo. Es decir, en caso de que se aplique la
45 BDPCM, la transformada puede saltarse y la cuantificación puede aplicarse con respecto a la muestra residual. Además, el dominio residual cuantificado puede incluir el coeficiente de transformada cuantificado. Un indicador que representa si la BDPCM es aplicable puede señalizarse a un nivel de secuencia (SPS), y dicho indicador puede señalizarse solo en caso de señalizarse de que el modo de salto de transformada es posible en el SPS. El indicador puede denominarse indicador habilitado para BDPCM o indicador habilitado para BDPCM de SPS.
50

En caso de aplicar la BDPCM, la intrapredicción puede llevarse a cabo con respecto a bloques completos mediante copia de muestra según una dirección de predicción (p. ej., predicción vertical o predicción horizontal) similar a la dirección de intrapredicción. El residuo que es el valor de diferencia entre los bloques original y de predicción puede cuantificarse mediante el salto de la transformada, y puede codificarse un valor delta, es decir,
55 el valor de diferencia entre el residuo cuantificado y el predictor en la dirección horizontal o vertical (es decir, el residuo cuantificado en la dirección horizontal o vertical).

Si la BDPCM es aplicable, el tamaño de la CU puede ser igual a o menor que MaxTsSize (tamaño máximo del bloque de salto de transformada) para la muestra de luma, y en caso de que la CU se codifique a través de intrapredicción, la información de indicador puede transmitirse al nivel de la CU. La información de indicador puede denominarse indicador BDPCM. En este caso, MaxTsSize puede significar el tamaño de bloque máximo para que se permita el modo de salto de transformada. La información de indicador puede indicar si se aplica intracodificación típica o se aplica la BDPCM. Si se aplica la BDPCM, puede transmitirse un indicador de dirección de predicción de BDPCM que indica si la dirección de predicción es la dirección horizontal o la
60 dirección vertical. El indicador de dirección de predicción de BDPCM puede denominarse indicador de dirección de BDPCM. A continuación, el bloque puede predecirse a través de un proceso de intrapredicción horizontal o
65

vertical típico usando una muestra de referencia no filtrada. Además, el residuo puede cuantificarse, y puede codificarse el valor de diferencia entre el residuo cuantificado y el predictor, por ejemplo, entre residuos ya cuantificados en posiciones circundantes en la dirección horizontal o vertical según la dirección de predicción de BDPCM.

5

Mientras tanto, como se describió anteriormente, la información (elemento sintáctico) en una tabla sintáctica descrita en el presente documento puede incluirse en la información de imagen/vídeo, y puede configurarse/codificarse por el aparato de codificación para transferirse al aparato de decodificación en forma de un flujo de bits. El aparato de decodificación puede analizar/decodificar la información (elemento sintáctico) en la tabla sintáctica correspondiente. El aparato de decodificación puede llevar a cabo un proceso de decodificación (predicción, proceso residual (basado en salto de transformada), BDPCM y codificación de paleta) para el bloque actual en base a la información decodificada.

10

En lo sucesivo, en el presente documento, se propone un esquema eficiente para analizar/señalar un elemento sintáctico que tiene dependencia con respecto a un elemento sintáctico de alto nivel relacionado con salto de transformada y/o elemento sintáctico de alto nivel relacionado con codificación de paleta. Es decir, según una realización del presente documento, durante la codificación de vídeo/imagen, si llevar a cabo la codificación puede clasificarse según la dependencia y no dependencia de la información que es absolutamente necesaria o auxiliar usada en la realización de la codificación de salto de transformada y/o paleta y, por lo tanto, puede llevarse a cabo una codificación eficiente.

15

20

En la codificación de vídeo, un conmutador de una herramienta de codificación puede definirse en una sintaxis de alto nivel (HLS, por sus siglas en inglés) específica. En el caso de la VVC en la técnica relacionada, la información de indicador sobre las herramientas de codificación respectivas puede definirse en el SPS. Además, en la VVC, la normalización ha estado en curso hacia tener independencia entre los respectivos conjuntos sintácticos de alto nivel (p. ej., conjunto de parámetros de vídeo (VPS), conjunto de parámetros de secuencia (SPS), conjunto de parámetros de imagen (PPS), conjunto de parámetros de adaptación (APS), conjunto de parámetros de decodificación (DPS, por sus siglas en inglés) y encabezado de segmento). Por consiguiente, en el conjunto de sintaxis de alto nivel en el que está presente un indicador de una herramienta de codificación, están presentes múltiples elementos sintácticos que tienen dependencia. En la(s) realización(es) del presente documento, se propone un método para analizar/señalar un elemento sintáctico de alto nivel que tiene dependencia según la codificación de paleta y/o salto de transformada.

25

30

Como una realización, el presente documento propone un método para guardar bits que se transmiten habilitando el elemento sintáctico que tiene dependencia para determinar si llevar a cabo análisis sintáctico/señalización según la condición dependiente con respecto al elemento sintáctico de alto nivel relacionado con salto de transformada. Como ejemplo, se propone un método para analizar sintácticamente el elemento sintáctico de alto nivel que tiene dependencia dependiendo de si usar el salto de transformada por el indicador de salto de transformada (habilitado).

35

40

Por ejemplo, como elementos sintácticos dependientes de la codificación basada en salto de transformada, hay un indicador de salto de transformada (habilitado) (p. ej., `sps_transform_skip_enabled_flag`), información de parámetro de cuantificación mínimo para el salto de transformada (p. ej., `min_qp_prime_ts_minus4`) e información sobre si aplicar la BDPCM (p. ej., `sps_bdpcm_enabled_flag`). Como ejemplo, si el valor del indicador de salto de transformada (habilitado) se define como 1, el indicador relacionado o los elementos sintácticos de información deben transmitirse necesariamente, mientras que si el valor del indicador de salto de transformada (habilitado) se define como 0, los elementos sintácticos que excluyen el elemento sintáctico del indicador de salto de transformada (habilitado) pueden no transmitirse.

45

Es decir, se propone un método para transmitir un elemento de sintaxis de alto nivel, que depende de si llevar a cabo el salto de transformada como, por ejemplo, la información de parámetro de cuantificación mínimo para el bloque de salto de transformada y si aplicar la BDPCM, durante el salto de transformada según el valor del indicador de salto de transformada (habilitado) en la HLS de sintaxis de alto nivel (p. ej., VPS, SPS, PPS, APS, DPS y encabezado de segmento). Además, el método propuesto no se limita a los elementos sintácticos descritos en la presente realización, sino que puede incluir todos los elementos sintácticos de alto nivel definidos en el conjunto sintáctico de alto nivel que tiene dependencia dependiendo de si llevar a cabo el salto de transformada y que incluye el indicador de salto de transformada (habilitado).

50

55

Como se ha descrito anteriormente, los elementos sintácticos relacionados con la codificación basada en el salto de transformada pueden definirse en el conjunto sintáctico de alto nivel, y pueden definirse en un conjunto de parámetros de secuencia (SPS) como en una realización en la Tabla 6 a continuación.

60

Tabla 6

	Descriptor
seq_parameter_set_rbsp() {	
(...)	
sps_transform_skip_enabled_flag	u(1)
si(sps_transform_skip_enabled_flag) {	
sps_bdpcm_enabled_flag	u(1)
min_qp_prime_ts_minus4	ue(v)
}	
(...)	
rbsp_trailing_bits()	
}	

5 Además, por ejemplo, la semántica de un elemento sintáctico para la realización descrita anteriormente entre elementos sintácticos de la sintaxis de SPS puede representarse como en la Tabla 7 a continuación.

Tabla 7

sps_transform_skip_enabled_flag igual a 1 especifica que transform_skip_flag puede estar presente en la sintaxis de unidad de transformada. sps_transform_skip_enabled_flag igual a 0 especifica que transform_skip_flag no está presente en la sintaxis de unidad de transformada.
sps_bdpcm_enabled_flag igual a 1 especifica que intra_bdpcm_flag puede estar presente en la sintaxis de unidad de codificación para unidades de intracodificación. sps_bdpcm_enabled_flag igual a 0 especifica que intra_bdpcm_flag no está presente en la sintaxis de unidad de codificación para unidades de intracodificación. Cuando no está presente, se infiere que el valor de sps_bdpcm_enabled_flag es igual a 0.
min_qp_prime_ts_minus4 especifica el parámetro de cuantificación mínimo permitido para el modo de salto de transformada de la siguiente manera:
$QpPrimeTsMin = 4 + min_qp_prime_ts_minus4$

10 Con referencia a la Tabla 6 y la Tabla 7 anteriores, los elementos sintácticos relacionados con el salto de transformada pueden definirse en el SPS y pueden incluir elementos sintácticos de sps_transform_skip_enabled_flag, sps_bdpcm_enabled_flag, y min_qp_prime_ts_minus4.

15 El elemento sintáctico sps_transform_skip_enabled_flag puede representar si el salto de transformada está habilitado basándose en si el valor del mismo es 0 o 1. Por ejemplo, si el valor de sps_transform_skip_enabled_flag es 1, puede representar que el salto de transformada está habilitado y, en este caso, transform_skip_flag puede analizarse/señalizarse a través de una sintaxis de unidad de transformada. En este caso, el elemento sintáctico transform_skip_flag puede representar si la transformada puede aplicarse al bloque de transformada asociado correspondiente. Si el valor de sps_transform_skip_enabled_flag es 0, puede representar que el salto de transformada no está habilitado y, en este caso, transform_skip_flag puede no analizarse/señalizarse en la sintaxis de unidad de transformada. En otras palabras, en base al indicador habilitado de salto de transformada sps_transform_skip_enabled_flag, puede representarse si transform_skip_flag está presente en la sintaxis de unidad de transformada.

25 El elemento sintáctico sps_bdpcm_enabled_flag puede representar si la BDPCM está habilitada basándose en si el valor del mismo es 0 o 1. Por ejemplo, si el valor de sps_bdpcm_enabled_flag es 1, puede representar que la BDPCM está habilitada y, en este caso, intra_bdpcm_flag (o intra_bdpcm_luma_flag e intra_bdpcm_chroma_flag) puede analizarse/señalizarse a través de la sintaxis de la unidad de codificación para la unidad de intracodificación. En este caso, el elemento sintáctico intra_bdpcm_flag puede representar si la BDPCM se aplica al bloque de codificación actual. Si el valor de sps_bdpcm_enabled_flag es 0, puede representar que la BDPCM no está habilitada y, en este caso, intra_bdpcm_flag (o intra_bdpcm_luma_flag e intra_bdpcm_chroma_flag) puede no analizarse/señalizarse en la sintaxis de la unidad de codificación para la unidad de intracodificación. En otras palabras, puede representarse si intra_bdpcm_flag (o intra_bdpcm_luma_flag e intra_bdpcm_chroma_flag) está presente en la sintaxis de unidad de codificación basándose en el indicador habilitado para BDPCM sps_bdpcm_enabled_flag.

35 El elemento sintáctico min_qp_prime_ts_minus4 puede representar el parámetro de cuantificación mínimo

5 permitido para el modo de salto de transformada. Por ejemplo, basándose en el elemento sintáctico min_qp_prime_ts_minus4, puede derivarse el valor de parámetro de cuantificación mínimo (p. ej., QpPrimeTsMin) en el modo de salto de transformada. Basándose en el parámetro de cuantificación mínimo en el modo de salto de transformada, puede derivarse el parámetro de cuantificación que se usa en un proceso de escalado (proceso de descuantificación). Además, un coeficiente de transformada escalado (coeficiente de transformada descuantificado) puede derivarse llevando a cabo el proceso de escalado (proceso de descuantificación) para el bloque actual basándose en el parámetro de cuantificación y, basándose en esto, puede derivarse una muestra residual del bloque actual.

10 Además, en el SPS, los elementos sintácticos que tienen dependencia pueden definirse con respecto al elemento sintáctico de indicador habilitado de salto de transformada (p. ej., sps_transform_skip_enabled_flag) entre los elementos sintácticos relacionados con el salto de transformada. Por ejemplo, como se describe en la Tabla 6 y la Tabla 7 anteriores, en base al valor del indicador habilitado de salto de transformada (p. ej., sps_transform_skip_enabled_flag) en el SPS, el elemento sintáctico min_qp_prime_ts_minus4 que representa la información de parámetro de cuantificación mínimo para el bloque de salto de transformada en el modo de salto de transformada y el elemento sintáctico sps_bdpcm_enabled_flag que representa si la BDPCM está habilitada puede tener dependencia. Como ejemplo, si el valor del indicador habilitado para salto de transformada (p. ej., sps_transform_skip_enabled_flag) es 1, el elemento sintáctico min_qp_prime_ts_minus4 y el elemento sintáctico sps_bdpcm_enabled_flag pueden analizarse/señalizarse. Además, si el valor del indicador habilitado de salto de transformada (p. ej., sps_transform_skip_enabled_flag) es 0, el elemento sintáctico min_qp_prime_ts_minus4 y el elemento sintáctico sps_bdpcm_enabled_flag pueden no analizarse/señalizarse.

15 20

25 Además, como una realización, el presente documento propone un método para guardar bits que se transmiten habilitando el elemento sintáctico que tiene dependencia para determinar si llevar a cabo el análisis sintáctico/señalización según la condición dependiente con respecto al elemento sintáctico de alto nivel relacionado con salto de transformada. Como ejemplo, se propone un método para analizar sintácticamente el elemento sintáctico de alto nivel que tiene dependencia dependiendo de si se usa el salto de transformada por el indicador de salto de transformada (habilitado).

30

35 Por ejemplo, como elementos sintácticos dependientes de la codificación basada en el salto de transformada, hay un indicador de salto de transformada (habilitado) (p. ej., sps_transform_skip_enabled_flag), información sobre el tamaño de aplicación de salto de transformada (p. ej., log2_transform_skip_max_size_minus2), información de parámetro de cuantificación mínimo para el salto de transformada (p. ej., min_qp_prime_ts_minus4), e información sobre si se aplica la BDPCM (p. ej., sps_bdpcm_enabled_flag). Como ejemplo, si el valor del indicador de salto de transformada (habilitado) se define como 1, el indicador relacionado o los elementos sintácticos de información deben transmitirse necesariamente, mientras que si el valor del indicador de salto de transformada (habilitado) se define como 0, los elementos sintácticos que excluyen el elemento sintáctico del indicador de salto de transformada (habilitado) pueden no transmitirse.

40

45 Es decir, se propone un método para transmitir un elemento de sintaxis de alto nivel, que depende de si llevar a cabo el salto de transformada como, por ejemplo, información sobre el tamaño máximo de la aplicación de salto de transformada, la información de parámetro de cuantificación mínimo y si aplicar la BDPCM durante el salto de transformada, según el valor del indicador de salto de transformada (habilitado) en la HLS de sintaxis de alto nivel (p. ej., VPS, SPS, PPS, APS, DPS y encabezado de segmento). Además, el método propuesto no se limita a los elementos sintácticos descritos en la presente realización, sino que puede incluir todos los elementos sintácticos de alto nivel definidos en el conjunto sintáctico de alto nivel que tiene dependencia dependiendo de si llevar a cabo el salto de transformada y que incluye el indicador de salto de transformada (habilitado).

50

55 Como se ha descrito anteriormente, los elementos sintácticos relacionados con la codificación basada en el salto de transformada pueden definirse en el conjunto sintáctico de alto nivel, y pueden definirse en el conjunto de parámetros de secuencia (SPS) como en una realización de la Tabla 8 a continuación. Sin embargo, la información de tamaño de bloque máximo para el salto de transformada que se define en un conjunto de parámetros de imagen (PPS) en la técnica relacionada puede definirse de manera reciente en el SPS para evitar la dependencia entre HLS, y esto puede representarse como en la Tabla 8 a continuación.

Tabla 8

seq_parameter_set_rbsp() {	Descriptor
(...)	
sps_transform_skip_enabled_flag	u(1)
si(sps_transform_skip_enabled_flag) {	
sps_bdpcm_enabled_flag	u(1)

	Descriptor
seq_parameter_set_rbsp() {	
min_qp_prime_ts_minus4	ue(v)
log2_transform_skip_max_size_minus2	ue(v)
}	
(...)	
rbsp_trailing_bits()	
}	

Además, por ejemplo, la semántica de un elemento sintáctico para la realización descrita anteriormente entre elementos sintácticos de la sintaxis de SPS puede representarse como en la Tabla 9 a continuación.

5

Tabla 9

sps_transform_skip_enabled_flag igual a 1 especifica que transform_skip_flag puede estar presente en la sintaxis de unidad de transformada. sps_transform_skip_enabled_flag igual a 0 especifica que transform_skip_flag no está presente en la sintaxis de unidad de transformada.
sps_bdpcm_enabled_flag igual a 1 especifica que intra_bdpcm_flag puede estar presente en la sintaxis de unidad de codificación para unidades de intracodificación. sps_bdpcm_enabled_flag igual a 0 especifica que intra_bdpcm_flag no está presente en la sintaxis de unidad de codificación para unidades de intracodificación. Cuando no está presente, se infiere que el valor de sps_bdpcm_enabled_flag es igual a 0.
min_qp_prime_ts_minus4 especifica el parámetro de cuantificación mínimo permitido para el modo de salto de transformada de la siguiente manera:
$QpPrimeTsMin = 4 + min_qp_prime_ts_minus4$
log2_transform_skip_max_size_minus2 especifica el tamaño máximo de bloque utilizado para el salto de transformada, y debe estar en el rango de 0 a 3.
Cuando no está presente, se infiere que el valor de log2_transform_skip_max_size_minus2 es igual a 0.
La variable MaxTsSize se establece igual a $1 \ll (\log2_transform_skip_max_size_minus2 + 2)$.

10 Con referencia a la Tabla 8 y la Tabla 9 anteriores, los elementos sintácticos relacionados con el salto de transformada pueden definirse en el SPS, y pueden incluir elementos sintácticos de sps_transform_skip_enabled_flag, sps_bdpcm_enabled_flag, min_qp_prime_ts_minus4, y log2_transform_skip_max_size_minus2.

15 En este caso, dado que los elementos sintácticos de sps_transform_skip_enabled_flag, sps_bdpcm_enabled_flag y min_qp_prime_ts_minus4 se han explicado en detalle en la Tabla 6 y la Tabla 7 anteriores, en la presente realización, la explicación detallada de los mismos se omitirá en aras de la explicación.

20 El elemento sintáctico log2_transform_skip_max_size_minus2 puede representar el tamaño máximo de bloque que se usa en el modo de salto de transformada. En este caso, el elemento de sintaxis log2_transform_skip_max_size_minus2 puede estar en el rango de 0 a 3. Por ejemplo, como se describe en la Tabla 9 anterior, el tamaño máximo de bloque (p. ej., MaxTsSize) que se usa en el modo de salto de transformada puede derivarse en base a cálculos como, por ejemplo, $1 \ll (\log2_transform_skip_max_size_minus2 + 2)$.

25 Además, en el SPS, los elementos sintácticos que tienen dependencia pueden definirse con respecto al elemento sintáctico de indicador habilitado de salto de transformada (p. ej., sps_transform_skip_enabled_flag) entre los elementos sintácticos relacionados con el salto de transformada. Por ejemplo, como se describe en la Tabla 8 y la Tabla 9 anteriores, en el SPS, el elemento sintáctico sps_bdpcm_enabled_flag que representa si la BDPCM está habilitada en base al valor del indicador habilitado de salto de transformada (p. ej., sps_transform_skip_enabled_flag), el elemento sintáctico min_qp_prime_ts_minus4 que representa la información de parámetro de cuantificación mínimo para el bloque de salto de transformada en el modo de salto de transformada, y el elemento sintáctico log2_transform_skip_max_size_minus2 que representa el tamaño de bloque máximo que se usa en el modo de salto de transformada pueden tener la dependencia. Como ejemplo, si el valor del indicador habilitado de salto de transformada (p. ej., sps_transform_skip_enabled_flag) es 1, los elementos sintácticos de sps_bdpcm_enabled_flag, min_qp_prime_ts_minus4, y
30 log2_transform_skip_max_size_minus2 pueden analizarse/señalizarse. Además, si el valor del indicador habilitado de salto de transformada (p. ej., sps_transform_skip_enabled_flag) es 0, los elementos sintácticos
35

de `sps_bdpcm_enabled_flag`, `min_qp_prime_ts_minus4`, y `log2_transform_skip_max_size_minus2` pueden no analizarse/señalizarse.

Además, como una realización, el presente documento propone un método para guardar bits que se transmiten habilitando el elemento sintáctico que tiene dependencia para determinar si llevar a cabo el análisis/señalización según la condición dependiente con respecto al elemento sintáctico de alto nivel relacionado con salto de transformada y el elemento sintáctico de alto nivel relacionado con codificación de paleta. Como ejemplo, se propone un método para analizar sintácticamente el elemento sintáctico de alto nivel que tiene dependencia mediante el indicador de salto de transformada (habilitado) y/o el indicador de codificación de paleta (habilitado).

Por ejemplo, como elementos sintácticos dependientes de la codificación basada en el salto de transformada, hay un indicador de salto de transformada (habilitado) (p. ej., `sps_transform_skip_enabled_flag`), información sobre el tamaño de aplicación de salto de transformada (p. ej., `log2_transform_skip_max_size_minus2`), información de parámetro de cuantificación mínimo para el salto de transformada (p. ej., `min_qp_prime_ts_minus4`) e información sobre si se aplica la BDPCM (p. ej., `sps_bdpcm_enabled_flag`). Además, como se describió anteriormente, dado que el valor de escape tampoco cambia durante la codificación de paleta, la información de parámetro de cuantificación mínimo para el salto de transformada puede usarse para llevar a cabo la cuantificación. Por consiguiente, el indicador de codificación de paleta (habilitado) (p. ej., `sps_palette_enabled_flag`) para la codificación basada en modo de paleta y la información de parámetro de cuantificación mínimo (p. ej., `min_qp_prime_ts_minus4`) durante el salto de transformada pueden tener dependencia. Como ejemplo, si el valor del indicador de salto de transformada (habilitado) o el indicador de codificación de paleta (habilitado) se define como 1, el indicador relacionado o los elementos sintácticos de información deben transmitirse necesariamente, mientras que si el valor del indicador de salto de transformada (habilitado) o el indicador de codificación de paleta (habilitado) se define como 0, los elementos sintácticos que excluyen el elemento sintáctico de indicador respectivo pueden no transmitirse.

Es decir, se propone un método para transmitir un elemento de sintaxis de alto nivel, que depende de si llevar a cabo el salto de transformada como, por ejemplo, información sobre el tamaño máximo de la aplicación de salto de transformada, la información de parámetro de cuantificación mínimo durante el salto de transformada, y si aplicar la BDPCM, o si llevar a cabo la codificación de paleta según el valor del indicador de salto de transformada (habilitado) y/o el valor del indicador de codificación de paleta (habilitado) en la sintaxis de alto nivel (p. ej., VPS, SPS, PPS, APS, DPS y encabezado de segmento).

Por ejemplo, (i) en caso de que tanto el indicador de salto de transformada (habilitado) como el indicador de codificación de paleta (habilitado) se definan como 1, los elementos sintácticos correspondientes a una unión de elementos sintácticos dependientes del indicador de salto de transformada (habilitado) y el indicador de codificación de paleta (habilitado) se puedan analizar. (ii) En caso de que el indicador de salto de transformada (habilitado) se defina como 1, y el indicador de codificación de paleta (habilitado) sea 0, los elementos sintácticos dependientes del indicador de salto de transformada (habilitado) pueden analizarse. (iii) En caso de que el indicador de salto de transformada (habilitado) se defina como 0, y el indicador de codificación de paleta (habilitado) sea 1, los elementos sintácticos dependientes del indicador de codificación de paleta (habilitado) se pueden analizar. (iv) En caso de que tanto el indicador de salto de transformada (habilitado) como el indicador de codificación de paleta (habilitado) sean 0, otros elementos sintácticos de alto nivel que tengan dependencia de dos herramientas de codificación pueden no ser analizados.

El orden de análisis de los elementos sintácticos descritos en la presente realización no está limitado específicamente, y en caso de que se determine si llevar a cabo el análisis según la dependencia entre los elementos sintácticos, se considera que coinciden entre sí. Además, el método propuesto no se limita a los elementos sintácticos descritos en la presente realización, sino que puede tener dependencia dependiendo de si se lleva a cabo el salto de transformada o la codificación de paleta, y puede incluir todos los elementos sintácticos de alto nivel definidos en el conjunto sintáctico de alto nivel que incluyen el indicador de salto de transformada (habilitado) y el indicador de codificación de paleta (habilitado).

Como se ha descrito anteriormente, los elementos sintácticos relacionados con la codificación basada en salto de transformada y/o la codificación basada en modo de paleta pueden definirse en el conjunto sintáctico de alto nivel, y pueden definirse en el conjunto de parámetros de secuencia (SPS) como en una realización de la Tabla 10 a continuación.

Tabla 10

seq_parameter_set_rbsp() {	Descriptor
(...)	
<code>sps_transform_skip_enabled_flag</code>	u(1)

seq_parameter_set_rbsp() {	Descriptor
si(sps_transform_skip_enabled_flag)	
sps_bdpcm_enabled_flag	u(1)
(...)	
si(chroma_format_idc == 3)	
sps_palette_enabled_flag	u(1)
(...)	
si(sps_transform_skip_enabled_flag sps_palette_enabled_flag)	
min_qp_prime_ts_minus4	ue(v)
(...)	
}	

Además, por ejemplo, la semántica del elemento sintáctico para la realización descrita anteriormente entre los elementos sintácticos de la sintaxis de SPS puede representarse como en la Tabla 11 a continuación.

5

Tabla 11

sps_transform_skip_enabled_flag igual a 1 especifica que transform_skip_flag puede estar presente en la sintaxis de unidad de transformada. sps_transform_skip_enabled_flag igual a 0 especifica que transform_skip_flag no está presente en la sintaxis de unidad de transformada.
sps_bdpcm_enabled_flag igual a 1 especifica que intra_bdpcm_flag puede estar presente en la sintaxis de unidad de codificación para unidades de intracodificación. sps_bdpcm_enabled_flag igual a 0 especifica que intra_bdpcm_flag no está presente en la sintaxis de unidad de codificación para unidades de intracodificación. Cuando no está presente, se infiere que el valor de sps_bdpcm_enabled_flag es igual a 0.
min_qp_prime_ts_minus4 especifica el parámetro de cuantificación mínimo permitido para el modo de salto de transformada de la siguiente manera: QpPrimeTsMin = 4 + min_qp_prime_ts_minus4
sps_palette_enabled_flag igual a 1 especifica que pred_mode_plt_flag puede estar presente en la sintaxis de unidad de codificación. sps_palette_enabled_flag igual a 0 especifica que pred_mode_plt_flag no está presente en la sintaxis de unidad de codificación. Cuando sps_palette_enabled_flag no está presente, se infiere que es igual a 0.

Con referencia a la Tabla 10 y la Tabla 11 anteriores, en el SPS, se pueden definir elementos sintácticos relacionados con el salto de transformada y/o la codificación de paleta, y pueden incluir elementos sintácticos de sps_transform_skip_enabled_flag, sps_bdpcm_enabled_flag, sps_palette_enabled_flag, y min_qp_prime_ts_minus4.

En este caso, dado que los elementos sintácticos de sps_transform_skip_enabled_flag, sps_bdpcm_enabled_flag, y min_qp_prime_ts_minus4 se han explicado en detalle en la Tabla 6 a la Tabla 9 anteriores, en la presente realización, en aras de la explicación, se omitirá la explicación detallada de los mismos.

El elemento sintáctico sps_palette_enabled_flag puede representar si la codificación de paleta (es decir, el modo de predicción de paleta) está habilitada basándose en si el valor del mismo es 0 o 1. Por ejemplo, si el valor de sps_palette_enabled_flag es 1, puede representar que la codificación de paleta está habilitada y, en este caso, pred_mode_plt_flag puede analizarse/señalizarse a través de la sintaxis de unidad de codificación. En este caso, el elemento sintáctico pred_mode_plt_flag puede representar si el modo de paleta puede usarse para la unidad de codificación actual. Si el valor de sps_palette_enabled_flag es 0, puede representar que la codificación de paleta no está habilitada y, en este caso, pred_mode_plt_flag puede no analizarse/señalizarse en la sintaxis de unidad de codificación. En otras palabras, en base a sps_palette_enabled_flag habilitado para la codificación de paleta, puede representar si pred_mode_plt_flag está presente en la sintaxis de unidad de codificación.

Además, en el SPS, los elementos sintácticos que tienen dependencia pueden definirse con respecto al elemento sintáctico de indicador habilitado de salto de transformada (p. ej., sps_transform_skip_enabled_flag) entre los elementos sintácticos relacionados con el salto de transformada y/o la codificación de paleta. Por ejemplo, como se describe en la Tabla 10 y la Tabla 11 anteriores, en el SPS, el elemento sintáctico

sps_bdpcm_enabled_flag que representa si la BDPCM está habilitada en base al valor del indicador de habilitación de salto de transformada (p. ej., sps_transform_skip_enabled_flag) puede tener dependencia. Como ejemplo, si el valor del indicador habilitado de salto de transformada (p. ej., sps_transform_skip_enabled_flag) es 1, el elemento sintáctico sps_bdpcm_enabled_flag puede analizarse/señalizarse. Además, si el valor del indicador habilitado de salto de transformada (p. ej., sps_transform_skip_enabled_flag) es 0, el elemento sintáctico sps_bdpcm_enabled_flag puede no analizarse/señalizarse.

Además, en el SPS, se puede definir una condición dependiente en relación con el elemento sintáctico de indicador habilitado de codificación de paleta (p. ej., sps_palette_enabled_flag) entre los elementos sintácticos relacionados con el salto de transformada y/o la codificación de paleta. Por ejemplo, como se describe en la Tabla 10 y la Tabla 11 anteriores, en el SPS, el elemento sintáctico de indicador habilitado de codificación de paleta (p. ej., sps_palette_enabled_flag) puede analizarse/señalizarse basándose en el elemento sintáctico chroma_format_idc. Como ejemplo, si el valor del elemento de sintaxis chroma_format_idc es 3, el elemento de sintaxis sps_palette_enabled_flag puede analizarse/señalizarse.

Además, en el SPS, los elementos sintácticos que tienen dependencia pueden definirse con respecto al elemento sintáctico de indicador habilitado de salto de transformada (p. ej., sps_transform_skip_enabled_flag) y/o el elemento sintáctico de indicador habilitado de codificación de paleta (p. ej., sps_palette_enabled_flag) entre los elementos sintácticos relacionados con el salto de transformada y/o la codificación de paleta. Por ejemplo, como se describe en la Tabla 10 y la Tabla 11 anteriores, en el SPS, el elemento sintáctico min_qp_prime_ts_minus4 que representa la información de parámetro de cuantificación mínimo para el modo de salto de transformada puede tener dependencia basándose en el valor del indicador habilitado de salto de transformada (p. ej., sps_transform_skip_enabled_flag) y/o el elemento sintáctico de indicador habilitado de codificación de paleta (p. ej., sps_palette_enabled_flag). Como ejemplo, en caso de que el valor del indicador habilitado de salto de transformada (p. ej., sps_transform_skip_enabled_flag) sea 1, o el valor del indicador habilitado de codificación de paleta (p. ej., sps_palette_enabled_flag) sea 1, el elemento sintáctico min_qp_prime_ts_minus4 puede analizarse/señalizarse.

Además, como se describió anteriormente, los elementos sintácticos relacionados con la codificación basada en salto de transformada y/o la codificación basada en modo de paleta pueden definirse en el conjunto sintáctico de alto nivel y, como en una realización de la Tabla 12 a continuación, pueden definirse en el conjunto de parámetros de secuencia (SPS). Sin embargo, la información de tamaño de bloque máximo para el salto de transformada que se define en un conjunto de parámetros de imagen (PPS) en la técnica relacionada puede definirse de manera reciente en el SPS para evitar la dependencia entre HLS, y esto puede representarse como en la Tabla 12 a continuación.

Tabla 12

seq_parameter_set_rbsp() {	Descriptor
(...)	
sps_transform_skip_enabled_flag	u(1)
si(sps_transform_skip_enabled_flag) {	
sps_bdpcm_enabled_flag	u(1)
log2_transform_skip_max_size_minus2	ue(v)
}	
(...)	
si(chroma_format_idc == 3)	
sps_palette_enabled_flag	u(1)
(...)	
si(sps_transform_skip_enabled_flag sps_palette_enabled_flag)	
min_qp_prime_ts_minus4	ue(v)
(...)	
}	

Además, por ejemplo, la semántica de un elemento sintáctico para la realización descrita anteriormente entre elementos sintácticos de la sintaxis de SPS puede representarse como en la Tabla 13 a continuación.

Tabla 13

sps_transform_skip_enabled_flag igual a 1 especifica que transform_skip_flag puede estar presente en la sintaxis de unidad de transformada. sps_transform_skip_enabled_flag igual a 0 especifica que transform_skip_flag no está presente en la sintaxis de unidad de transformada.
sps_bdpcm_enabled_flag igual a 1 especifica que intra_bdpcm_flag puede estar presente en la sintaxis de unidad de codificación para unidades de intracodificación. sps_bdpcm_enabled_flag igual a 0 especifica que intra_bdpcm_flag no está presente en la sintaxis de unidad de codificación para unidades de intracodificación. Cuando no está presente, se infiere que el valor de sps_bdpcm_enabled_flag es igual a 0.
min_qp_prime_ts_minus4 especifica el parámetro de cuantificación mínimo permitido para el modo de salto de transformada de la siguiente manera: QpPrimeTsMin = 4 + min_qp_prime_ts_minus4
log2_transform_skip_max_size_minus2 especifica el tamaño máximo de bloque utilizado para el salto de transformada, y debe estar en el rango de 0 a 3. Cuando no está presente, se infiere que el valor de log2_transform_skip_max_size_minus2 es igual a 0.
La variable MaxTsSize se establece igual a $1 \ll (\log_2 \text{transform_skip_max_size_minus2} + 2)$.
sps_palette_enabled_flag igual a 1 especifica que pred_mode_plt_flag puede estar presente en la sintaxis de unidad de codificación. sps_palette_enabled_flag igual a 0 especifica que pred_mode_plt_flag no está presente en la sintaxis de unidad de codificación. Cuando sps_palette_enabled_flag no está presente, se infiere que es igual a 0.

5 Con referencia a la Tabla 12 y la Tabla 13 anteriores, en el SPS, se pueden definir elementos sintácticos relacionados con el salto de transformada y/o la codificación de paleta, y pueden incluir elementos sintácticos de sps_transform_skip_enabled_flag, sps_bdpcm_enabled_flag, log2_transform_skip_max_size_minus2, sps_palette_enabled_flag y min_qp_prime_ts_minus4.

10 En este caso, dado que los elementos sintácticos de sps_transform_skip_enabled_flag, sps_bdpcm_enabled_flag, log2_transform_skip_max_size_minus2, sps_palette_enabled_flag y min_qp_prime_ts_minus4 se han descrito en detalle en la Tabla 6 a la Tabla 11 anteriores, en la presente realización, en aras de la explicación, se omitirá la explicación detallada de los mismos.

15 Como se describe en una realización de la Tabla 12 y la Tabla 13 anteriores, en el SPS, los elementos sintácticos que tienen dependencia pueden definirse con respecto al elemento sintáctico de indicador habilitado de salto de transformada (p. ej., sps_transform_skip_enabled_flag) entre los elementos sintácticos relacionados con el salto de transformada y/o la codificación de paleta. Por ejemplo, como se describe en la Tabla 12 y la Tabla 13 anteriores, en el SPS, el elemento sintáctico sps_bdpcm_enabled_flag que representa si la BDPCM está habilitada en base al valor del indicador habilitado de salto de transformada (p. ej., sps_transform_skip_enabled_flag), y el elemento sintáctico log2_transform_skip_max_size_minus2 que representa el tamaño de bloque máximo que se usa en el modo de salto de transformada puede tener la dependencia. Como ejemplo, si el valor del indicador habilitado de salto de transformada (p. ej., sps_transform_skip_enabled_flag) es 1, los elementos sintácticos de sps_bdpcm_enabled_flag y log2_transform_skip_max_size_minus2 pueden analizarse/señalizarse. Además, si el valor del indicador de habilitación de salto de transformada (p. ej., sps_transform_skip_enabled_flag) es 0, los elementos sintácticos de sps_bdpcm_enabled_flag y log2_transform_skip_max_size_minus2 pueden no analizarse/señalizarse.

25 Además, en el SPS, los elementos sintácticos que tienen dependencia pueden definirse con respecto al elemento sintáctico de indicador habilitado de salto de transformada (p. ej., sps_transform_skip_enabled_flag) y/o el elemento sintáctico de indicador habilitado de codificación de paleta (p. ej., sps_palette_enabled_flag) entre los elementos sintácticos relacionados con el salto de transformada y/o la codificación de paleta. Por ejemplo, como se describe en la Tabla 12 y la Tabla 13 anteriores, el elemento sintáctico min_qp_prime_ts_minus4 que representa la información de parámetro de cuantificación mínimo para el modo de salto de transformada basándose en los valores del indicador habilitado de salto de transformada (p. ej., sps_transform_skip_enabled_flag) y/o el elemento sintáctico de indicador habilitado de codificación de paleta (p. ej., sps_palette_enabled_flag) puede tener la dependencia en el SPS. Como ejemplo, en caso de que el valor del indicador habilitado de salto de transformada (p. ej., sps_transform_skip_enabled_flag) sea 1, o el valor del indicador habilitado de codificación de paleta (p. ej., sps_palette_enabled_flag) sea 1, el elemento sintáctico min_qp_prime_ts_minus4 puede analizarse/señalizarse.

40 Mientras tanto, la fuente o la imagen codificada puede incluir una matriz de componentes de luma y, en algunos casos, puede incluir además dos matrices de componentes de croma (cb, cr), es decir, un píxel de la imagen puede incluir una muestra de luma y una muestra de croma (cb, cr).

5 Un formato de color puede representar un formato de configuración del componente de luma y el componente de croma (cb, cr), y puede denominarse un formato de croma. El formato de color (o formato de croma) puede estar predeterminado, o puede señalizarse de manera adaptativa. Por ejemplo, el formato de croma puede señalizarse basándose en al menos uno de `chroma_format_idc` y `separate_colour_plane_flag` como en la Tabla 14 a continuación.

Tabla 14

<code>chroma_format_idc</code>	<code>separate_colour_plane_flag</code>	Formato de croma	SubWidthC	SubHeightC
0	0	Monocromo	1	1
1	0	4:2:0	2	2
2	0	4:2:2	2	1
3	0	4:4:4	1	1
3	1	4:4:4	1	1

10 Con referencia a la Tabla 14 anterior, en el muestreo monocromático, solo hay una matriz de muestras que se considera nominalmente como una matriz de luma.

15 En el muestreo 4:2:0, cada una de las dos matrices de croma tiene un media altura y un medio ancho de la matriz de luma.

En el muestreo 4:2:2, cada una de las dos matrices de croma tiene la misma altura y la mitad del ancho de la matriz de luma.

20 En el muestreo 4:4:4, se pueden aplicar los siguientes según el valor de `separate_colour_plane_flag`.

- Si el valor de `separate_colour_plane_flag` es 0, cada una de las dos matrices de croma tiene la misma altura y el mismo ancho de la matriz de luma.

25 - De lo contrario, si el valor de `separate_colour_plane_flag` es 1, se pueden procesar por separado tres tipos de planos de color como imágenes muestreadas monocromáticas.

30 `SubWidthC` y `SubHeightC` pueden representar una relación entre la muestra de luma y la muestra de croma. Por ejemplo, si `chroma_format_idc` es 3, el formato de croma es 4:4:4 y, en este caso, en caso de que el ancho del bloque de muestra de luma sea 16, el ancho del bloque de muestra de croma correspondiente puede ser $16/\text{SubWidthC}$. En general, la sintaxis relacionada con la muestra de croma y el flujo de bits pueden analizarse solo en caso de que el tipo de matriz de croma (p. ej., `chromaArrayType`) no sea 0.

35 Además, como una realización, el presente documento propone un método para guardar bits que se transmiten habilitando el elemento sintáctico que tiene dependencia para determinar si llevar a cabo el análisis/señalización según la condición dependiente con respecto al elemento sintáctico de alto nivel relacionado con salto de transformada y el elemento sintáctico de alto nivel relacionado con codificación de paleta. Como ejemplo, se propone un método para analizar sintácticamente el elemento sintáctico de alto nivel que tiene dependencia por el indicador de salto de transformada (habilitado) y el indicador de codificación de paleta (habilitado).

40 Por ejemplo, como elementos sintácticos dependientes de la codificación basada en el salto de transformada, hay un indicador de salto de transformada (habilitado) (p. ej., `sps_transform_skip_enabled_flag`), información sobre el tamaño de la aplicación de salto de transformada (p. ej., `log2_transform_skip_max_size_minus2`), información de parámetro de cuantificación mínimo durante el salto de transformada (p. ej., `min_qp_prime_ts_minus4`) e información sobre si aplicar la BDPCM (p. ej., `sps_bdpcm_enabled_flag`). Además, como se describió anteriormente, dado que el valor de escape tampoco cambia durante la codificación de paleta, la información de parámetro de cuantificación mínimo para el salto de transformada puede usarse para llevar a cabo la cuantificación.

50 Como se representa en la realización descrita anteriormente, si el valor del indicador de salto de transformada (habilitado) o el indicador de codificación de paleta (habilitado) se define como 1, el indicador relacionado o los elementos sintácticos de información deben transmitirse necesariamente, mientras que si el valor del indicador de salto de transformada (habilitado) o el indicador de codificación de paleta (habilitado) se define como 0, los elementos sintácticos que excluyen los elementos sintácticos de indicador respectivos pueden no transmitirse.

55 Es decir, se propone un método para transmitir un elemento sintáctico de alto nivel, que depende de si llevar a cabo el salto de transformada como, por ejemplo, la información de parámetro de cuantificación mínimo durante

el salto de transformada o la codificación de paleta y si aplicar la BDPCM, según el valor del indicador de salto de transformada (habilitado) y el indicador de codificación de paleta (habilitado), o si llevar a cabo la codificación de paleta en la sintaxis de alto nivel (p. ej., VPS, SPS, PPS, APS, DPS y encabezado de segmento).

5 Por ejemplo, (i) En caso de que tanto el indicador de salto de transformada (habilitado) como el indicador de codificación de paleta (habilitado) se definan como 1, los elementos sintácticos correspondientes a una unión de elementos sintácticos dependientes del indicador de salto de transformada (habilitado) y el indicador de codificación de paleta (habilitado) se puedan analizar. (ii) En caso de que el indicador de salto de transformada (habilitado) se defina como 1, y el indicador de codificación de paleta (habilitado) sea 0, los elementos sintácticos dependientes del indicador de salto de transformada (habilitado) pueden analizarse. (iii) En caso de que el indicador de salto de transformada (habilitado) se defina como 0, y el indicador de codificación de paleta (habilitado) sea 1, los elementos sintácticos dependientes del indicador de codificación de paleta (habilitado) se pueden analizar. (iv) En caso de que tanto el indicador de salto de transformada (habilitado) como el indicador de codificación de paleta (habilitado) sean 0, otros elementos sintácticos de alto nivel que tengan dependencia de dos herramientas de codificación pueden no ser analizados.

El orden de análisis de los elementos sintácticos descritos en la presente realización no está limitado específicamente, y en caso de que se determine si llevar a cabo el análisis según la dependencia entre los elementos sintácticos, se considera que coinciden entre sí. Además, el método propuesto no se limita a los elementos sintácticos descritos en la presente realización, sino que puede tener dependencia dependiendo de si se lleva a cabo el salto de transformada o la codificación de paleta, y puede incluir todos los elementos sintácticos de alto nivel definidos en el conjunto sintáctico de alto nivel que incluyen el indicador de salto de transformada (habilitado) y el indicador de codificación de paleta (habilitado).

25 Como se ha descrito anteriormente, los elementos sintácticos relacionados con la codificación basada en salto de transformada y/o la codificación basada en modo de paleta pueden definirse en el conjunto sintáctico de alto nivel, y pueden definirse en el conjunto de parámetros de secuencia (SPS) como en una realización de la Tabla 15 a continuación.

30 Tabla 15

seq_parameter_set_rbsp() {	Descriptor
(...)	
sps_transform_skip_enabled_flag	u(1)
si(sps_transform_skip_enabled_flag)	
sps_bdpcm_enabled_flag	u(1)
(...)	
si(chroma_format_idc == 3)	
sps_palette_enabled_flag	u(1)
(...)	
si(sps_palette_enabled_flag)	
min_qp_prime_ts_chroma_minus4	ue(v)
(...)	
si(sps_transform_skip_enabled_flag sps_palette_enabled_flag)	
min_qp_prime_ts_luma_minus4	ue(v)
(...)	
}	

Además, por ejemplo, la semántica del elemento sintáctico para la realización descrita anteriormente entre los elementos sintácticos de la sintaxis de SPS puede representarse como en la Tabla 16 a continuación.

35

Tabla 16

sps_transform_skip_enabled_flag igual a 1 especifica que transform_skip_flag puede estar presente en la sintaxis de unidad de transformada. sps_transform_skip_enabled_flag igual a 0 especifica que transform_skip_flag no está presente en la sintaxis de unidad de transformada.
sps_bdpcm_enabled_flag igual a 1 especifica que intra_bdpcm_flag puede estar presente en la sintaxis de unidad de codificación para unidades de intracodificación. sps_bdpcm_enabled_flag igual a 0 especifica que intra_bdpcm_flag no está presente en la sintaxis de unidad de codificación para unidades de intracodificación. Cuando no está presente, se infiere que el valor de sps_bdpcm_enabled_flag es igual a 0.
min_qp_prime_ts_luma_minus4 especifica el parámetro de cuantificación mínimo permitido para el modo de salto de transformada en el componente de luma de la siguiente manera: QpPrimeTsMin = 4 + min_qp_prime_ts_luma_minus4
min_qp_prime_ts_chroma_minus4 especifica el parámetro de cuantificación mínimo permitido para el modo de salto de transformada en el componente de croma de la siguiente manera: QpPrimeTsMin = 4 + min_qp_prime_ts_chroma_minus4
sps_palette_enabled_flag igual a 1 especifica que pred_mode_plt_flag puede estar presente en la sintaxis de unidad de codificación. sps_palette_enabled_flag igual a 0 especifica que pred_mode_plt_flag no está presente en la sintaxis de unidad de codificación. Cuando sps_palette_enabled_flag no está presente, se infiere que es igual a 0.

5 Con referencia a la Tabla 15 y la Tabla 16 anteriores, en el SPS, se pueden definir elementos sintácticos relacionados con el salto de transformada y/o la codificación de paleta, y pueden incluir elementos sintácticos de sps_transform_skip_enabled_flag, sps_bdpcm_enabled_flag, sps_palette_enabled_flag, min_qp_prime_ts_luma_minus4, y min_qp_prime_ts_chroma_minus4.

10 En este caso, dado que los elementos sintácticos de sps_transform_skip_enabled_flag, sps_bdpcm_enabled_flag, y sps_palette_enabled_flag se han explicado en detalle en la Tabla 6 a la Tabla 11 anteriores, en la presente realización, en aras de la explicación, se omitirá la explicación detallada de los mismos.

15 Como se describe en una realización de la Tabla 15 y la Tabla 16 anteriores, en el SPS, los elementos sintácticos que tienen dependencia pueden definirse con respecto al elemento sintáctico de indicador habilitado de salto de transformada (p. ej., sps_transform_skip_enabled_flag) entre los elementos sintácticos relacionados con el salto de transformada y/o la codificación de paleta. Por ejemplo, como se describe en la Tabla 15 y la Tabla 16 anteriores, en el SPS, el elemento sintáctico sps_bdpcm_enabled_flag que representa si la BDPCM está habilitada en base al valor del indicador de habilitación de salto de transformada (p. ej., sps_transform_skip_enabled_flag) puede tener la dependencia. Como ejemplo, si el valor del indicador habilitado de salto de transformada (p. ej., sps_transform_skip_enabled_flag) es 1, el elemento sintáctico sps_bdpcm_enabled_flag puede analizarse/señalizarse. Además, si el valor del indicador habilitado de salto de transformada (p. ej., sps_transform_skip_enabled_flag) es 0, el elemento sintáctico sps_bdpcm_enabled_flag puede no analizarse/señalizarse.

25 Además, en el SPS, los elementos sintácticos que tienen dependencia pueden definirse con respecto al elemento sintáctico de indicador habilitado de codificación de paleta (p. ej., sps_palette_enabled_flag) entre los elementos sintácticos relacionados con el salto de transformada y/o la codificación de paleta. Por ejemplo, como se describe en la Tabla 15 y la Tabla 16 anteriores, el elemento sintáctico min_qp_prime_ts_chroma_minus4 que representa la información de parámetro de cuantificación mínimo en el modo de salto de transformada para el componente de croma basándose en el valor del indicador habilitado de codificación de paleta (p. ej., sps_palette_enabled_flag) puede tener la dependencia en el SPS. Como un ejemplo, si el valor del indicador habilitado de codificación de paleta (p. ej., sps_palette_enabled_flag) es 1, el elemento sintáctico min_qp_prime_ts_chroma_minus4 puede analizarse/señalizarse. Además, si el valor del indicador habilitado de codificación de paleta (p. ej., sps_palette_enabled_flag) es 0, el elemento sintáctico min_qp_prime_ts_chroma_minus4 puede no analizarse/señalizarse.

40 Además, en el SPS, los elementos sintácticos que tienen dependencia pueden definirse con respecto al elemento sintáctico de indicador habilitado de salto de transformada (p. ej., sps_transform_skip_enabled_flag) y/o el elemento sintáctico de indicador habilitado de codificación de paleta (p. ej., sps_palette_enabled_flag) entre los elementos sintácticos relacionados con el salto de transformada y/o la codificación de paleta. Por ejemplo, como se describe en la Tabla 15 y la Tabla 16 anteriores, el elemento sintáctico min_qp_prime_ts_luma_minus4 que representa la información de parámetro de cuantificación mínimo en el modo de salto de transformada para el componente de luma basándose en el valor del indicador habilitado de salto de transformada (p. ej., sps_transform_skip_enabled_flag) y/o el elemento sintáctico de indicador habilitado de codificación de paleta (p. ej., sps_palette_enabled_flag) puede tener la dependencia en el SPS.

Como ejemplo, si el valor del indicador habilitado de salto de transformada (p. ej., `sps_transform_skip_enabled_flag`) es 1, o el valor del indicador habilitado de codificación de paleta (p. ej., `sps_palette_enabled_flag`) es 1, el elemento sintáctico `min_qp_prime_ts_luma_minus4` puede analizarse/señalizarse.

5

Además, como una realización, el presente documento propone un método para guardar bits que se transmiten habilitando el elemento sintáctico que tiene dependencia para determinar si llevar a cabo el análisis/señalización según la condición dependiente con respecto al elemento sintáctico de alto nivel relacionado con el salto de transformada y el elemento sintáctico de alto nivel relacionado con codificación de paleta. Como ejemplo, se propone un método para analizar sintácticamente el elemento sintáctico de alto nivel que tiene dependencia por el indicador de salto de transformada (habilitado) y el indicador de codificación de paleta (habilitado).

10

Por ejemplo, como elementos sintácticos dependientes de la codificación basada en el salto de transformada, hay un indicador de salto de transformada (habilitado) (p. ej., `sps_transform_skip_enabled_flag`), información sobre el tamaño de la aplicación de salto de transformada (p. ej., `log2_transform_skip_max_size_minus2`), información de parámetro de cuantificación mínimo durante el salto de transformada (p. ej., `min_qp_prime_ts_minus4`) e información sobre si aplicar la BDPCM (p. ej., `sps_bdpcm_enabled_flag`). Además, como se describió anteriormente, dado que el valor de escape tampoco cambia durante la codificación de paleta, la información de parámetro de cuantificación mínimo para el salto de transformada puede usarse para llevar a cabo la cuantificación.

15

20

Como se representa en la realización descrita anteriormente, si el valor del indicador de salto de transformada (habilitado) o el indicador de codificación de paleta (habilitado) se define como 1, el indicador relacionado o los elementos sintácticos de información deben transmitirse necesariamente, mientras que si el valor del indicador de salto de transformada (habilitado) o el indicador de codificación de paleta (habilitado) se define como 0, los elementos sintácticos que excluyen los elementos sintácticos de indicador respectivos pueden no transmitirse. Es decir, se propone un método para transmitir un elemento de sintaxis de alto nivel, que depende de si llevar a cabo el salto de transformada como, por ejemplo, información sobre el tamaño máximo de la aplicación de salto de transformada, la información de parámetro de cuantificación mínimo durante el salto de transformada o la codificación de paleta, y si aplicar la BDPCM, según el valor del indicador de salto de transformada (habilitado) y el indicador de codificación de paleta (habilitado), o si llevar a cabo la codificación de paleta en la sintaxis de alto nivel (p. ej., VPS, SPS, PPS, APS, DPS y encabezado de segmento).

25

30

Por ejemplo, (i) En caso de que tanto el indicador de salto de transformada (habilitado) como el indicador de codificación de paleta (habilitado) se definan como 1, los elementos sintácticos correspondientes a una unión de elementos sintácticos dependientes del indicador de salto de transformada (habilitado) y el indicador de codificación de paleta (habilitado) se puedan analizar. (ii) En caso de que el indicador de salto de transformada (habilitado) se defina como 1, y el indicador de codificación de paleta (habilitado) sea 0, los elementos sintácticos dependientes del indicador de salto de transformada (habilitado) pueden analizarse. (iii) En caso de que el indicador de salto de transformada (habilitado) se defina como 0, y el indicador de codificación de paleta (habilitado) sea 1, los elementos sintácticos dependientes del indicador de codificación de paleta (habilitado) se pueden analizar. (iv) En caso de que tanto el indicador de salto de transformada (habilitado) como el indicador de codificación de paleta (habilitado) sean 0, otros elementos sintácticos de alto nivel que tengan dependencia de dos herramientas de codificación pueden no ser analizados.

35

40

45

El orden de análisis de los elementos sintácticos descritos en la presente realización no está limitado específicamente, y en caso de que se determine si llevar a cabo el análisis según la dependencia entre los elementos sintácticos, se considera que coinciden entre sí. Además, el método propuesto no se limita a los elementos sintácticos descritos en la presente realización, sino que puede tener dependencia dependiendo de si se lleva a cabo el salto de transformada o la codificación de paleta, y puede incluir todos los elementos sintácticos de alto nivel definidos en el conjunto sintáctico de alto nivel que incluyen el indicador de salto de transformada (habilitado) y el indicador de codificación de paleta (habilitado).

50

Como se ha descrito anteriormente, los elementos sintácticos relacionados con la codificación basada en salto de transformada y/o la codificación basada en modo de paleta pueden definirse en el conjunto sintáctico de alto nivel, y pueden definirse en el conjunto de parámetros de secuencia (SPS) como en una realización de la Tabla 17 a continuación. En la presente realización, la información sobre el tamaño máximo del salto de transformada definido en el conjunto de parámetros de imagen (PPS) en la técnica relacionada puede definirse de manera reciente en el SPS para evitar la dependencia entre las HLS, y se propone un método para llevar a cabo el análisis/señalización basándose en la dependencia del salto de transformada y el elemento sintáctico relacionado con la codificación de paleta que se ha usado anteriormente.

55

60

Tabla 17

seq_parameter_set_rbsp() {	Descriptor
(...)	
sps_transform_skip_enabled_flag	u(1)
si(sps_transform_skip_enabled_flag) {	
sps_bdpcm_enabled_flag	u(1)
log2_transform_skip_max_size_minus2	ue(v)
}	
(...)	
si(chroma_format_idc == 3)	
sps_palette_enabled_flag	u(1)
(...)	
si(sps_palette_enabled_flag)	
min_qp_prime_ts_chroma_minus4	ue(v)
(...)	
si(sps_transform_skip_enabled_flag sps_palette_enabled_flag)	
min_qp_prime_ts_luma_minus4	ue(v)
(...)	
}	

5 Además, por ejemplo, la semántica del elemento sintáctico para la realización descrita anteriormente entre los elementos sintácticos de la sintaxis de SPS puede representarse como en la Tabla 18 a continuación.

Tabla 18

sps_transform_skip_enabled_flag igual a 1 especifica que transform_skip_flag puede estar presente en la sintaxis de unidad de transformada. sps_transform_skip_enabled_flag igual a 0 especifica que transform_skip_flag no está presente en la sintaxis de unidad de transformada.
sps_bdpcm_enabled_flag igual a 1 especifica que intra_bdpcm_flag puede estar presente en la sintaxis de unidad de codificación para unidades de intracodificación. sps_bdpcm_enabled_flag igual a 0 especifica que intra_bdpcm_flag no está presente en la sintaxis de unidad de codificación para unidades de intracodificación. Cuando no está presente, se infiere que el valor de sps_bdpcm_enabled_flag es igual a 0.
log2_transform_skip_max_size_minus2 especifica el tamaño máximo de bloque utilizado para el salto de transformada, y debe estar en el rango de 0 a 3.
Cuando no está presente, se infiere que el valor de log2_transform_skip_max_size_minus2 es igual a 0.
La variable MaxTsSize se establece igual a $1 \ll (\log_2 \text{transform_skip_max_size_minus2} + 2)$.
min_qp_prime_ts_luma_minus4 especifica el parámetro de cuantificación mínimo permitido para el modo de salto de transformada en el componente de luma de la siguiente manera:
$\text{QpPrimeTsMin} = 4 + \text{min_qp_prime_ts_luma_minus4}$
min_qp_prime_ts_chroma_minus4 especifica el parámetro de cuantificación mínimo permitido para el modo de salto de transformada en el componente de croma de la siguiente manera:
$\text{QpPrimeTsMin} = 4 + \text{min_qp_prime_ts_chroma_minus4}$
sps_palette_enabled_flag igual a 1 especifica que pred_mode_plt_flag puede estar presente en la sintaxis de unidad de codificación. sps_palette_enabled_flag igual a 0 especifica que pred_mode_plt_flag no está presente en la sintaxis de unidad de codificación. Cuando sps_palette_enabled_flag no está presente, se infiere que es igual a 0.

10 Con referencia a la Tabla 17 y la Tabla 18 anteriores, en el SPS, se pueden definir elementos sintácticos

relacionados con el salto de transformada y/o la codificación de paleta, y pueden incluir elementos sintácticos de `sps_transform_skip_enabled_flag`, `sps_bdpcm_enabled_flag`, `log2_transform_skip_max_size_minus2`, `sps_palette_enabled_flag`, `min_qp_prime_ts_luma_minus4`, y `min_qp_prime_ts_chroma_minus4`.

5 En este caso, dado que los elementos sintácticos de `sps_transform_skip_enabled_flag`, `sps_bdpcm_enabled_flag`, `log2_transform_skip_max_size_minus2`, `sps_palette_enabled_flag`, `min_qp_prime_ts_luma_minus4`, y `min_qp_prime_ts_chroma_minus4` se han explicado en detalle en la Tabla 6 a la Tabla 11 anteriores, en la presente realización, en aras de la explicación, se omitirá la explicación detallada de los mismos.

10 Como se describe en una realización de la Tabla 17 y la Tabla 18 anteriores, en el SPS, los elementos sintácticos que tienen dependencia pueden definirse con respecto al elemento sintáctico de indicador habilitado de salto de transformada (p. ej., `sps_transform_skip_enabled_flag`) entre los elementos sintácticos relacionados con el salto de transformada y/o la codificación de paleta. Por ejemplo, como se describe en la
 15 Tabla 17 y la Tabla 18 anteriores, en el SPS, el elemento sintáctico `sps_bdpcm_enabled_flag` que representa si la BDPCM está habilitada en base al valor del indicador habilitado de salto de transformada (p. ej., `sps_transform_skip_enabled_flag`) y el elemento sintáctico `log2_transform_skip_max_size_minus2` que representa el tamaño de bloque máximo que se usa en el modo de salto de transformada puede tener la dependencia. Como ejemplo, si el valor del indicador habilitado de salto de transformada (p. ej., `sps_transform_skip_enabled_flag`) es 1, los elementos sintácticos de `sps_bdpcm_enabled_flag` y `log2_transform_skip_max_size_minus2` pueden analizarse/señalizarse. Además, si el valor del indicador de habilitación de salto de transformada (p. ej., `sps_transform_skip_enabled_flag`) es 0, los elementos sintácticos de `sps_bdpcm_enabled_flag` y `log2_transform_skip_max_size_minus2` pueden no analizarse/señalizarse. Además, en el SPS, los elementos sintácticos que tienen dependencia pueden definirse con respecto al
 25 elemento sintáctico de indicador habilitado de codificación de paleta (p. ej., `sps_palette_enabled_flag`) entre los elementos sintácticos relacionados con el salto de transformada y/o la codificación de paleta. Por ejemplo, como se describe en la Tabla 17 y la Tabla 18 anteriores, el elemento sintáctico `min_qp_prime_ts_chroma_minus4` que representa la información de parámetro de cuantificación mínimo en el modo de salto de transformada para el componente de croma basándose en el valor del indicador habilitado de codificación de paleta (p. ej., `sps_palette_enabled_flag`) puede tener la dependencia en el SPS. Como un
 30 ejemplo, si el valor del indicador habilitado de codificación de paleta (p. ej., `sps_palette_enabled_flag`) es 1, el elemento sintáctico `min_qp_prime_ts_chroma_minus4` puede analizarse/señalizarse. Además, si el valor del indicador habilitado de codificación de paleta (p. ej., `sps_palette_enabled_flag`) es 0, el elemento sintáctico `min_qp_prime_ts_chroma_minus4` puede no analizarse/señalizarse.

35 Además, en el SPS, los elementos sintácticos que tienen dependencia pueden definirse con respecto al elemento sintáctico de indicador habilitado de salto de transformada (p. ej., `sps_transform_skip_enabled_flag`) y/o el elemento sintáctico de indicador habilitado de codificación de paleta (p. ej., `sps_palette_enabled_flag`) entre los elementos sintácticos relacionados con el salto de transformada y/o la codificación de paleta. Por
 40 ejemplo, como se describe en la Tabla 17 y la Tabla 18 anteriores, el elemento sintáctico `min_qp_prime_ts_luma_minus4` que representa la información de parámetro de cuantificación mínimo en el modo de salto de transformada para el componente de luma basándose en el valor del indicador habilitado de salto de transformada (p. ej., `sps_transform_skip_enabled_flag`) y/o el elemento sintáctico de indicador habilitado de codificación de paleta (p. ej., `sps_palette_enabled_flag`) puede tener la dependencia en el SPS. Como ejemplo, si el valor del indicador habilitado de salto de transformada (p. ej., `sps_transform_skip_enabled_flag`) es 1, o el valor del indicador habilitado de codificación de paleta (p. ej., `sps_palette_enabled_flag`) es 1, el elemento sintáctico `min_qp_prime_ts_luma_minus4` puede
 45 analizarse/señalizarse.

50 El siguiente dibujo se ha preparado para explicar un ejemplo detallado del presente documento. Dado que el nombre de un dispositivo detallado o un término o nombre detallado (p. ej., sintaxis/nombre del elemento sintáctico) descrito en el dibujo se presenta a modo de ejemplo, las características técnicas del presente documento no se limitan al nombre detallado usado en el dibujo. Las FIGS. 10 y 11 ilustran esquemáticamente un método de codificación de vídeo/imagen y un ejemplo de componentes relacionados según la(s)
 55 realización(es) del presente documento.

Las FIGS. 10 y 11 ilustran esquemáticamente un método de codificación de vídeo/imagen y un ejemplo de componentes relacionados según la(s) realización(es) del presente documento.

60 El método descrito en la FIG. 10 puede llevarse a cabo por el aparato 200 de codificación descrito en la FIG. 2 o la FIG. 11. Aquí, el aparato 200 de codificación descrito en la FIG. 11 representa brevemente el aparato 200 de codificación descrito en la FIG. 2. Específicamente, las etapas E1000 a E1010 de la FIG. 10 pueden llevarse a cabo por el procesador 230 residual descrito en la FIG. 2, y la etapa E1020 de la FIG. 10 puede llevarse a
 65 cabo por el codificador 240 de entropía descrito en la FIG. 2. Además, aunque no se ilustra, el predictor 220 del aparato 200 de codificación puede llevar a cabo un proceso de derivación de una muestra de predicción, el

sumador 250 del aparato 200 de codificación puede llevar a cabo un proceso de generación de una muestra reconstruida para el bloque actual y una imagen reconstruida basándose en la muestra residual y la muestra de predicción para el bloque actual, y el codificador 240 de entropía del aparato 200 de codificación puede llevar a cabo un proceso de codificación de información de predicción para el bloque actual. Además, el método descrito en la FIG. 10 puede incluir las realizaciones descritas anteriormente del presente documento a llevar a cabo. Por consiguiente, con referencia a la FIG. 10, se omitirá o simplificará la explicación detallada de los contenidos duplicados de las realizaciones descritas anteriormente.

Con referencia a la FIG. 10, el aparato de codificación determina si aplicar la transformada al bloque actual en base a la información habilitada de salto de transformada (E1000).

Como una realización, el aparato de codificación puede determinar en primer lugar el modo de predicción para el bloque actual, y puede derivar muestras de predicción. Por ejemplo, el aparato de codificación puede determinar si llevar a cabo interpredicción o intrapredicción con respecto al bloque actual, y también puede determinar un modo de interpredicción específico o un modo de intrapredicción específico basándose en un coste de RD. Además, el aparato de codificación puede determinar si llevar a cabo la predicción con respecto al bloque actual basándose en un modo CIIP, un modo IBC, un modo BDPCM o un modo de paleta. El aparato de codificación puede derivar las muestras de predicción para el bloque actual llevando a cabo la predicción según el modo de predicción determinado. En este caso, se pueden aplicar varios métodos de predicción descritos en el presente documento como, por ejemplo, interpredicción o intrapredicción. Además, el aparato de codificación puede generar y codificar información (p. ej., información de modo de predicción) relacionada con la predicción aplicada al bloque actual.

Además, el aparato de codificación puede derivar las muestras residuales comparando las muestras de predicción con las muestras originales para el bloque actual entre sí. El aparato de codificación puede derivar coeficientes de transformada a través de un proceso de transformada para las muestras residuales. En este caso, el aparato de codificación puede determinar si aplicar la transformada al bloque actual en consideración con la eficiencia de codificación. Es decir, el aparato de codificación puede determinar si la transformada se aplica a las muestras residuales del bloque actual.

Por ejemplo, el aparato de codificación determina si aplicar la transformada o el modo de salto de transformada al bloque actual (muestras residuales) en base a la información habilitada de salto de transformada.

Como se ha descrito anteriormente, la información habilitada de salto de transformada es información sobre si el salto de transformada está habilitado y, como se describe en la Tabla 6 a la Tabla 18, puede representarse como el elemento sintáctico `sps_transform_skip_enabled_flag`. Por ejemplo, si el valor de `sps_transform_skip_enabled_flag` es 1, representa que el salto de transformada está habilitado y, en este caso, `transform_skip_flag` se analiza/señaliza a través de la sintaxis de unidad de transformada. En este caso, el elemento sintáctico `transform_skip_flag` representa si la transformada puede aplicarse al bloque de transformada asociado. Si el valor de `sps_transform_skip_enabled_flag` es 0, representa que el salto de transformada no está habilitado y, en este caso, `transform_skip_flag` no se analiza/señaliza en la sintaxis de unidad de transformada. La información habilitada de salto de transformada (p. ej., `sps_transform_skip_enabled_flag`) puede incluirse en el SPS y puede señalizarse al aparato de decodificación. Es decir, basándose en que el valor de la información habilitada de salto de transformada (p. ej., `sps_transform_skip_enabled_flag`) incluida en el SPS es 1, la sintaxis de la unidad de transformada puede incluir el indicador de salto de transformada (p. ej., `transform_skip_flag`). En este caso, si el valor del indicador de salto de transformada (p. ej., `transform_skip_flag`) incluido en la sintaxis de unidad de transformada es 1, puede llevarse a cabo un modo en el que no se aplica la transformada (modo de salto de transformada) para el bloque actual. Además, si el valor del indicador de salto de transformada (p. ej., `transform_skip_flag`) incluido en la sintaxis de unidad de transformada es 0, la transformada se aplica al bloque actual.

Por ejemplo, si el valor de la información habilitada de salto de transformada es 1 (es decir, con respecto a la información habilitada de salto de transformada que representa que el salto de transformada está habilitado), el aparato de codificación determina si aplicar la transformada al bloque actual. Es decir, el aparato de codificación genera información (indicador de salto de transformada) sobre si aplicar la transformada al bloque actual basándose en que el valor de la información habilitada de salto de transformada es 1, y señala el indicador de salto de transformada a través de la sintaxis de unidad de transformada. En este caso, si la transformada no se aplica al bloque actual (es decir, en el caso del modo de salto de transformada), el aparato de codificación genera un indicador de salto de transformada cuyo valor es 1, e incluye esto en la sintaxis de unidad de transformada. Además, en caso de aplicar la transformada al bloque actual, el aparato de codificación genera el indicador de salto de transformada cuyo valor es 0, e incluye esto en la sintaxis de unidad de transformada.

El aparato de codificación genera información residual para el bloque actual basándose en si aplicar la transformada (E1010).

Como una realización, el aparato de codificación deriva muestras residuales del bloque actual, y genera información residual aplicando la transformada o salto de transformada a las muestras residuales del bloque

actual en base a si aplicar la transformada. Por ejemplo, con respecto a las muestras residuales para el bloque actual cuyo valor de indicador de salto de transformada es 1, el aparato de codificación aplica el modo de salto de transformada. En este caso, el aparato de codificación deriva las muestras residuales del bloque actual como coeficientes de transformada. Además, con respecto a las muestras residuales para el bloque actual cuyo valor de indicador de salto de transformada es 0, el aparato de codificación deriva los coeficientes de transformada llevando a cabo la transformada. El aparato de codificación puede derivar coeficientes de transformada cuantificados llevando a cabo un proceso de cuantificación basado en los coeficientes de transformada derivados a través del salto de transformada o la transformada. El aparato de codificación puede generar la información residual basándose en los coeficientes de transformada cuantificados.

En este caso, la información residual puede ser información generada a través del proceso de transformada y/o cuantificación, y puede ser información sobre los coeficientes de transformada cuantificados y, por ejemplo, puede incluir información sobre información de valor de los coeficientes de transformada cuantificados, información de posición, técnica de transformada, núcleo de transformada y parámetro de cuantificación.

El aparato de codificación codifica información de imagen (o información de vídeo) (E1020).

Aquí, la información de imagen puede incluir la información residual. Además, la información de imagen puede incluir información (p. ej., información de modo de predicción) relacionada con la predicción usada para derivar las muestras de predicción. Además, la información de imagen puede incluir información relacionada con el salto de transformada, por ejemplo, información habilitada de salto de transformada e información de indicador de salto de transformada. Es decir, la información de imagen puede incluir varios tipos de información derivada en el proceso de codificación, y la información de imagen que incluye dichos varios tipos de información puede codificarse.

Además, la información de imagen puede incluir diversos tipos de información según la(s) realización(es) descrita(s) anteriormente en el presente documento, y puede incluir información descrita en al menos una de las Tablas 1 a 18 tal como se describió anteriormente.

Por ejemplo, la información de imagen puede incluir un conjunto de parámetros de secuencia (SPS). El SPS puede incluir información relacionada con el salto de transformada e información relacionada con la codificación de paleta. Como ejemplo, la información relacionada con el salto de transformada puede incluir información habilitada de salto de transformada (p. ej., `sps_transform_skip_enabled_flag`), información habilitada de BDPCM (p. ej., `sps_bdpcm_enabled_flag`), información sobre el tamaño de bloque máximo usado en el modo de salto de transformada (p. ej., `log2_transform_skip_max_size_minus2`) y la información de parámetro de cuantificación mínimo relacionada con el parámetro de cuantificación mínimo permitido para el modo de salto de transformada (p. ej., `min_qp_prime_ts_minus4`). Además, como ejemplo, la información relacionada con la codificación de paleta puede incluir información habilitada de la codificación de paleta (p. ej., `sps_palette_enabled_flag`) y la información de parámetro de cuantificación mínimo relacionada con el parámetro de cuantificación mínimo permitido para el modo de salto de transformada (p. ej., `min_qp_prime_ts_minus4`).

Además, por ejemplo, como se describió anteriormente, entre la información relacionada con el salto de transformada y/o la codificación de paleta incluida en el SPS, la información que tiene dependencia puede definirse con respecto a la información habilitada de salto de transformada (p. ej., `sps_transform_skip_enabled_flag`).

Como ejemplo, el SPS puede configurarse para analizar/señalizar información de indicador habilitada de BDPCM (p. ej., `sps_bdpcm_enabled_flag`) sobre si habilitar la BDPCM basándose en el valor de la información habilitada de salto de transformada (p. ej., `sps_transform_skip_enabled_flag`). En este caso, si el valor de la información de indicador habilitado de salto de transformada (p. ej., `sps_transform_skip_enabled_flag`) es 1, la información de indicador habilitado de BDPCM (p. ej., `sps_bdpcm_enabled_flag`) puede incluirse en el SPS, y la información (p. ej., `sps_bdpcm_enabled_flag`) puede analizarse/señalizarse desde el SPS. Además, si el valor de la información de indicador habilitado de salto de transformada (p. ej., `sps_transform_skip_enabled_flag`) es 0, la información de indicador habilitado de BDPCM (p. ej., `sps_bdpcm_enabled_flag`) puede no analizarse/señalizarse desde el SPS.

Además, basándose en el valor de la información de indicador habilitada de BDPCM (p. ej., `sps_bdpcm_enabled_flag`) en el SPS, la información de indicador de BDPCM (p. ej., `intra_bdpcm_flag`) sobre si aplicar la BDPCM al bloque actual) puede analizarse/señalizarse a través de la sintaxis de unidad de codificación. En este caso, si el valor de la información de indicador habilitada de BDPCM (p. ej., `sps_bdpcm_enabled_flag`) es 1, la información de indicador de BDPCM (p. ej., `intra_bdpcm_flag`) puede incluirse en la sintaxis de unidad de codificación, y la información (p. ej., `intra_bdpcm_flag`) puede analizarse/señalizarse a partir de la sintaxis de unidad de codificación. Además, si el valor de la información de indicador habilitada de BDPCM (p. ej., `sps_bdpcm_enabled_flag`) es 0, la información de indicador de BDPCM (p. ej., `intra_bdpcm_flag`) puede no analizarse/señalizarse a partir de la sintaxis de unidad de codificación.

Además, como ejemplo, el SPS puede configurarse para analizar/señalizar información sobre el tamaño de bloque máximo (p. ej., `log2_transform_skip_max_size_minus2`) usado en el modo de salto de transformada basándose en el valor de la información habilitada de salto de transformada (p. ej., `sps_transform_skip_enabled_flag`). En este caso, si el valor de la información de indicador habilitado de salto de transformada (p. ej., `sps_transform_skip_enabled_flag`) es 1, la información sobre el tamaño de bloque máximo (p. ej., `log2_transform_skip_max_size_minus2`) usada en el modo de salto de transformada puede incluirse en el SPS, y la información (p. ej., `log2_transform_skip_max_size_minus2`) puede analizarse/señalizarse desde el SPS. Además, si el valor de la información de indicador habilitado de salto de transformada (p. ej., `sps_transform_skip_enabled_flag`) es 0, la información sobre el tamaño de bloque máximo (p. ej., `log2_transform_skip_max_size_minus2`) usada en el modo de salto de transformada puede no analizarse/señalizarse desde el SPS.

Además, como ejemplo, basándose en el valor de la información de indicador habilitado de salto de transformada (p. ej., `sps_transform_skip_enabled_flag`) definida en el SPS, la información de indicador de salto de transformada (p. ej., `transform_skip_flag`) sobre si aplicar el salto de transformada al bloque actual puede analizarse/señalizarse a través de la sintaxis de unidad de transformada. En este caso, si el valor de la información de indicador habilitado de salto de transformada (p. ej., `sps_transform_skip_enabled_flag`) es 1, la información de indicador de salto de transformada (p. ej., `transform_skip_flag`) puede incluirse en la sintaxis de unidad de transformada, y la información (p. ej., `transform_skip_flag`) puede analizarse/señalizarse a partir de la sintaxis de unidad de transformada. Además, si el valor de la información de indicador habilitado de salto de transformada (p. ej., `sps_transform_skip_enabled_flag`) es 0, la información de indicador de salto de transformada (p. ej., `transform_skip_flag`) puede no analizarse/señalizarse a partir de la sintaxis de unidad de transformada.

Además, por ejemplo, entre la información relacionada con el salto de transformada y/o la codificación de paleta incluida en el SPS como se describió anteriormente, la información que tiene dependencia puede definirse con respecto a la información habilitada de codificación de paleta sobre si la codificación de paleta está habilitada (p. ej., `sps_palette_enabled_flag`). Como ejemplo, basándose en el valor de la información habilitada de codificación de paleta (p. ej., `sps_palette_enabled_flag`) definida en el SPS, la información de indicador de modo de predicción de paleta (p. ej., `pred_mode_plt_flag`) sobre si aplicar la codificación de paleta (modo de predicción de paleta) al bloque actual puede analizarse/señalizarse a través de la sintaxis de unidad de codificación. En este caso, si el valor de la información habilitada de codificación de paleta (p. ej., `sps_palette_enabled_flag`) es 1, la información de indicador de modo de predicción de paleta (p. ej., `pred_mode_plt_flag`) puede incluirse en la sintaxis de unidad de codificación, y la información (p. ej., `pred_mode_plt_flag`) puede analizarse/señalizarse a partir de la sintaxis de unidad de codificación. Además, si el valor de la información habilitada de codificación de paleta (p. ej., `sps_palette_enabled_flag`) es 0, la información de indicador de modo de predicción de paleta (p. ej., `pred_mode_plt_flag`) puede no analizarse/señalizarse a partir de la sintaxis de unidad de codificación.

Además, por ejemplo, como se describió anteriormente, entre la información relacionada con el salto de transformada y/o la codificación de paleta incluida en el SPS, la información que tiene dependencia puede definirse con respecto a la información habilitada de salto de transformada (p. ej., `sps_transform_skip_enabled_flag`) y/o la información habilitada de codificación de paleta (p. ej., `sps_palette_enabled_flag`).

Como ejemplo, en el SPS, en base a al menos una de la información habilitada de salto de transformada (p. ej., `sps_transform_skip_enabled_flag`) y/o la información habilitada de codificación de paleta (p. ej., `sps_palette_enabled_flag`), la información de parámetro de cuantificación mínimo (p. ej., `min_qp_prime_ts_minus4`) relacionada con el parámetro de cuantificación mínimo permitido para el modo de salto de transformada se analiza/señaliza. En otras palabras, basándose en la condición de que el valor de la información habilitada de salto de transformada (p. ej., `sps_transform_skip_enabled_flag`) sea 1, o el valor de la información habilitada de codificación de paleta (p. ej., `sps_palette_enabled_flag`) sea 1, la información de parámetro de cuantificación mínimo (p. ej., `min_qp_prime_ts_minus4`) puede incluirse en el SPS, y solo en caso de que se satisfaga la condición anterior, la información de parámetro de cuantificación mínimo (p. ej., `min_qp_prime_ts_minus4`) se analiza/señaliza.

En este caso, tal como se describió anteriormente, la información de parámetro de cuantificación mínimo (p. ej., `min_qp_prime_ts_minus4`) es información relacionada con el parámetro de cuantificación mínimo permitido para el modo de salto de transformada y, basándose en esto, se deriva el parámetro de cuantificación para el bloque actual.

Por ejemplo, en caso de que el modo de salto de transformada se aplique al bloque actual, el parámetro de cuantificación para el bloque actual se deriva en base a la información de parámetro de cuantificación mínimo (p. ej., `min_qp_prime_ts_minus4`), y los coeficientes de transformada cuantificados pueden derivarse llevando a cabo el proceso de cuantificación en base al parámetro de cuantificación.

Además, por ejemplo, en caso de que el modo de codificación de paleta se aplique al bloque actual, el

parámetro de cuantificación para un valor de escape del bloque actual se deriva basándose en la información de parámetro de cuantificación mínimo (p. ej., `min_qp_prime_ts_minus4`). En este caso, el valor de escape cuantificado (p. ej., `palette_escape_val`) puede derivarse aplicando el parámetro de cuantificación al valor de escape del bloque actual basándose en el parámetro de cuantificación. El proceso en el que se aplica el modo de codificación de paleta puede llevarse a cabo como se describe en la Tabla 4 y la Tabla 5 anteriores.

Como se ha descrito anteriormente, la información de imagen que incluye varios tipos de información puede codificarse y emitirse en forma de un flujo de bits. El flujo de bits puede transmitirse al aparato de decodificación a través de una red o un medio de almacenamiento (digital). En este caso, la red puede incluir una red de radiodifusión y/o una red de comunicación, y el medio de almacenamiento digital puede incluir diversos medios de almacenamiento como, por ejemplo, USB, SD, CD, DVD, Blu-ray, HDD y SSD.

Las FIGS. 12 y 13 ilustran esquemáticamente un método de decodificación de vídeo/imagen y un ejemplo de componentes relacionados según la(s) realización(es) del presente documento.

El método descrito en la FIG. 12 puede llevarse a cabo por el aparato 300 de decodificación descrito en la FIG. 3 o la FIG. 13. Aquí, el aparato 300 de decodificación descrito en la FIG. 13 representa brevemente el aparato 300 de decodificación descrito en la FIG. 3. Específicamente, la etapa E1200 de la FIG. 12 puede llevarse a cabo por el decodificador 310 de entropía descrito en la FIG. 3, las etapas E1210 a E1220 de la FIG. 12 pueden llevarse a cabo por el procesador 320 residual descrito en la FIG. 3, y la etapa E1230 de la FIG. 12 puede llevarse a cabo por el sumador 340 descrito en la FIG. 3. Además, aunque no se ilustra, el decodificador 310 de entropía del aparato 300 de decodificación puede llevar a cabo un proceso de recepción de información de predicción para el bloque actual, y el predictor 330 del aparato 300 de decodificación puede llevar a cabo un proceso de derivación de una muestra de predicción del bloque actual. Además, el método descrito en la FIG. 12 puede incluir las realizaciones descritas anteriormente del presente documento a llevar a cabo. Por consiguiente, con referencia a la FIG. 12, se omitirá o simplificará la explicación detallada de los contenidos duplicados de las realizaciones descritas anteriormente.

Con referencia a la FIG. 12, el aparato de decodificación recibe información de imagen (o información de vídeo) de un flujo de bits (E1200).

Como una realización, el aparato de decodificación deriva información (p. ej., información de vídeo/imagen) necesaria para la reconstrucción de imágenes analizando sintácticamente el flujo de bits. En este caso, la información de imagen incluye información residual, y la información residual puede incluir información de valor de coeficientes de transformada cuantificados, información de ubicación, técnica de transformada, núcleo de transformada e información de parámetros de cuantificación. Además, la información de imagen puede incluir información relacionada con la predicción (p. ej., información del modo de predicción). Además, la información de imagen incluye información relacionada con el salto de transformada, por ejemplo, información habilitada de salto de transformada e información de indicador de salto de transformada. Es decir, la información de imagen puede incluir varios tipos de información necesaria en el proceso de decodificación, y puede decodificarse basándose en un método de codificación como, por ejemplo, la codificación de Golomb exponencial, CAVLC o CABAC.

Además, la información de imagen puede incluir varios tipos de información según las realizaciones descritas anteriormente del presente documento, y puede incluir información descrita en al menos una de las Tablas 1 a 18 como se ha descrito anteriormente.

Por ejemplo, la información de imagen puede incluir un conjunto de parámetros de secuencia (SPS). El SPS puede incluir información relacionada con el salto de transformada e información relacionada con la codificación de paleta. Como ejemplo, la información relacionada con el salto de transformada puede incluir información habilitada para el salto de transformada (p. ej., `sps_transform_skip_enabled_flag`), información habilitada para BDPCM (p. ej., `sps_bdpcm_enabled_flag`), información sobre el tamaño de bloque máximo usado en el modo de salto de transformada (p. ej., `log2_transform_skip_max_size_minus2`) y la información de parámetro de cuantificación mínimo relacionada con el parámetro de cuantificación mínimo permitido para el modo de salto de transformada (p. ej., `min_qp_prime_ts_minus4`). Además, como ejemplo, la información relacionada con la codificación de paleta puede incluir información habilitada para la codificación de paleta (p. ej., `sps_palette_enabled_flag`) y la información de parámetro de cuantificación mínimo relacionada con el parámetro de cuantificación mínimo permitido para el modo de salto de transformada (p. ej., `min_qp_prime_ts_minus4`).

Además, por ejemplo, como se describió anteriormente, entre la información relacionada con el salto de transformada y/o la codificación de paleta incluida en el SPS, la información que tiene dependencia puede definirse con respecto a la información habilitada para salto de transformada (p. ej., `sps_transform_skip_enabled_flag`).

Como ejemplo, el SPS puede configurarse para analizar/señalar información de indicador habilitada de

- BDPCM (p. ej., `sps_bdpcm_enabled_flag`) sobre si habilitar la BDPCM basándose en el valor de la información habilitada de salto de transformada (p. ej., `sps_transform_skip_enabled_flag`). En este caso, si el valor de la información de indicador habilitado de salto de transformada (p. ej., `sps_transform_skip_enabled_flag`) es 1, la información de indicador habilitado de BDPCM (p. ej., `sps_bdpcm_enabled_flag`) puede incluirse en el SPS, y la información (p. ej., `sps_bdpcm_enabled_flag`) puede analizarse/señalizarse desde el SPS. Además, si el valor de la información de indicador habilitado de salto de transformada (p. ej., `sps_transform_skip_enabled_flag`) es 0, la información de indicador habilitado de BDPCM (p. ej., `sps_bdpcm_enabled_flag`) puede no analizarse/señalizarse desde el SPS.
- Además, basándose en el valor de la información de indicador habilitada de BDPCM (p. ej., `sps_bdpcm_enabled_flag`) en el SPS, la información de indicador de BDPCM (p. ej., `intra_bdpcm_flag`) sobre si aplicar la BDPCM al bloque actual) puede analizarse/señalizarse a través de la sintaxis de unidad de codificación. En este caso, si el valor de la información de indicador habilitada de BDPCM (p. ej., `sps_bdpcm_enabled_flag`) es 1, la información de indicador de BDPCM (p. ej., `intra_bdpcm_flag`) puede incluirse en la sintaxis de unidad de codificación, y la información (p. ej., `intra_bdpcm_flag`) puede analizarse/señalizarse a partir de la sintaxis de unidad de codificación. Además, si el valor de la información de indicador habilitada de BDPCM (p. ej., `sps_bdpcm_enabled_flag`) es 0, la información de indicador de BDPCM (p. ej., `intra_bdpcm_flag`) puede no analizarse/señalizarse a partir de la sintaxis de unidad de codificación.
- Además, como ejemplo, el SPS puede configurarse para analizar/señalizar información sobre el tamaño máximo de bloque (p. ej., `log2_transform_skip_max_size_minus2`) usado en el modo de salto de transformada basándose en el valor de la información habilitada de salto de transformada (p. ej., `sps_transform_skip_enabled_flag`). En este caso, si el valor de la información de indicador habilitado de salto de transformada (p. ej., `sps_transform_skip_enabled_flag`) es 1, la información sobre el tamaño de bloque máximo (p. ej., `log2_transform_skip_max_size_minus2`) usada en el modo de salto de transformada puede incluirse en el SPS, y la información (p. ej., `log2_transform_skip_max_size_minus2`) puede analizarse/señalizarse desde el SPS. Además, si el valor de la información de indicador habilitado de salto de transformada (p. ej., `sps_transform_skip_enabled_flag`) es 0, la información sobre el tamaño de bloque máximo (p. ej., `log2_transform_skip_max_size_minus2`) usada en el modo de salto de transformada puede no analizarse/señalizarse desde el SPS.
- Además, como ejemplo, basándose en el valor de la información de indicador habilitado de salto de transformada (p. ej., `sps_transform_skip_enabled_flag`) definida en el SPS, la información de indicador de salto de transformada (p. ej., `transform_skip_flag`) sobre si aplicar el salto de transformada al bloque actual puede analizarse/señalizarse a través de la sintaxis de unidad de transformada. En este caso, si el valor de la información de indicador habilitado de salto de transformada (p. ej., `sps_transform_skip_enabled_flag`) es 1, la información de indicador de salto de transformada (p. ej., `transform_skip_flag`) puede incluirse en la sintaxis de unidad de transformada, y la información (p. ej., `transform_skip_flag`) puede analizarse/señalizarse a partir de la sintaxis de unidad de transformada. Además, si el valor de la información de indicador habilitado de salto de transformada (p. ej., `sps_transform_skip_enabled_flag`) es 0, la información de indicador de salto de transformada (p. ej., `transform_skip_flag`) puede no analizarse/señalizarse a partir de la sintaxis de unidad de transformada.
- Además, por ejemplo, entre la información relacionada con el salto de transformada y/o la codificación de paleta incluida en el SPS como se describió anteriormente, la información que tiene dependencia puede definirse con respecto a la información habilitada para codificación de paleta sobre si la codificación de paleta está habilitada (p. ej., `sps_palette_enabled_flag`). Como ejemplo, basándose en el valor de la información habilitada de codificación de paleta (p. ej., `sps_palette_enabled_flag`) definida en el SPS, la información de indicador de modo de predicción de paleta (p. ej., `pred_mode_plt_flag`) sobre si aplicar la codificación de paleta (modo de predicción de paleta) al bloque actual puede analizarse/señalizarse a través de la sintaxis de unidad de codificación. En este caso, si el valor de la información habilitada de codificación de paleta (p. ej., `sps_palette_enabled_flag`) es 1, la información de indicador de modo de predicción de paleta (p. ej., `pred_mode_plt_flag`) puede incluirse en la sintaxis de unidad de codificación, y la información (p. ej., `pred_mode_plt_flag`) puede analizarse/señalizarse a partir de la sintaxis de unidad de codificación. Además, si el valor de la información habilitada de codificación de paleta (p. ej., `sps_palette_enabled_flag`) es 0, la información de indicador de modo de predicción de paleta (p. ej., `pred_mode_plt_flag`) puede no analizarse/señalizarse a partir de la sintaxis de unidad de codificación.
- Además, por ejemplo, como se describió anteriormente, entre la información relacionada con el salto de transformada y/o la codificación de paleta incluida en el SPS, la información que tiene dependencia puede definirse con respecto a la información habilitada para salto de transformada (p. ej., `sps_transform_skip_enabled_flag`) y/o la información habilitada para codificación de paleta (p. ej., `sps_palette_enabled_flag`).
- Como ejemplo, en el SPS, en base a al menos una de la información habilitada para salto de transformada (p. ej., `sps_transform_skip_enabled_flag`) y/o la información habilitada para la codificación de paleta (p. ej., `sps_palette_enabled_flag`), la información de parámetro de cuantificación mínimo (p. ej.,

min_qp_prime_ts_minus4) relacionada con el parámetro de cuantificación mínimo permitido para el modo de salto de transformada puede analizarse/señalizarse. En otras palabras, en base a la condición de que el valor de la información habilitada para el salto de transformada (p. ej., sps_transform_skip_enabled_flag) sea 1, o el valor de la información habilitada para la codificación de paleta (p. ej., sps_palette_enabled_flag) sea 1, la información de parámetro de cuantificación mínimo (p. ej., min_qp_prime_ts_minus4) puede incluirse en el SPS, y solo en caso de que se satisfaga la condición anterior, la información de parámetro de cuantificación mínimo (p. ej., min_qp_prime_ts_minus4) puede analizarse/señalizarse.

En este caso, tal como se describió anteriormente, la información de parámetro de cuantificación mínimo (p. ej., min_qp_prime_ts_minus4) puede ser información relacionada con el parámetro de cuantificación mínimo permitido para el modo de salto de transformada y, basándose en esto, puede derivarse el parámetro de cuantificación para el bloque actual.

Por ejemplo, en caso de que el modo de salto de transformada se aplique al bloque actual, el parámetro de cuantificación para el bloque actual puede derivarse basándose en la información de parámetro de cuantificación mínimo (p. ej., min_qp_prime_ts_minus4) y el coeficiente de transformada descuantificado (coeficiente de transformada escalado) puede derivarse llevando a cabo un proceso de descuantificación (proceso de escalado) basándose en el parámetro de cuantificación. Basándose en el coeficiente de transformada descuantificado, puede derivarse la muestra residual del bloque actual.

Además, por ejemplo, en caso de que el modo de codificación de paleta se aplique al bloque actual, el parámetro de cuantificación para un valor de escape del bloque actual puede derivarse basándose en la información de parámetro de cuantificación mínimo (p. ej., min_qp_prime_ts_minus4). En este caso, el valor de escape del bloque actual puede derivarse llevando a cabo la descuantificación (proceso de escalado) basándose en el parámetro de cuantificación. En base al valor de escape, se puede generar la muestra reconstruida del bloque actual. El proceso en el que se aplica el modo de codificación de paleta puede llevarse a cabo como se describe en la Tabla 4 y la Tabla 5 anteriores.

El aparato de decodificación determina si aplicar la transformada al bloque actual basándose en la información habilitada para el salto de transformada (E1210).

Como una realización, si se recibe la información de imagen que incluye la información habilitada para el salto de transformada, el aparato de decodificación determina si aplicar la transformada o el modo de salto de transformada al bloque actual basándose en la información habilitada de salto de transformada.

Como se ha descrito anteriormente, la información habilitada para el salto de transformada es información sobre si el salto de transformada está habilitado, y puede representarse como el elemento sintáctico sps_transform_skip_enabled_flag como se describe en la Tabla 6 a la Tabla 18. Por ejemplo, si el valor de sps_transform_skip_enabled_flag es 1, puede representar que el salto de transformada está habilitado y, en este caso, transform_skip_flag puede analizarse/señalizarse a través de la sintaxis de unidad de transformada. En este caso, el elemento sintáctico transform_skip_flag puede representar si la transformada puede aplicarse al bloque de transformada asociado. Si el valor de sps_transform_skip_enabled_flag es 0, puede representar que el salto de transformada no está habilitado y, en este caso, transform_skip_flag puede no analizarse/señalizarse en la sintaxis de unidad de transformada. La información habilitada para el salto de transformada (p. ej., sps_transform_skip_enabled_flag) puede incluirse en el SPS, y puede señalizarse del aparato de codificación al aparato de decodificación. Es decir, basándose en que el valor de la información habilitada para el salto de transformada (p. ej., sps_transform_skip_enabled_flag) incluida en el SPS es 1, la sintaxis de la unidad de transformada puede incluir el indicador de salto de transformada (p. ej., transform_skip_flag). En este caso, si el valor del indicador de salto de transformada (p. ej., transform_skip_flag) incluido en la sintaxis de unidad de transformada es 1, puede llevarse a cabo un modo en el que no se aplica la transformada (modo de salto de transformada) para el bloque actual. Además, si el valor del indicador de salto de transformada (p. ej., transform_skip_flag) incluido en la sintaxis de unidad de transformada es 0, la transformada puede aplicarse para el bloque actual.

Por ejemplo, si el valor de la información habilitada para el salto de transformada es 1 (es decir, para la información habilitada para el salto de transformada que representa que el salto de transformada está habilitado), el aparato de decodificación determina si aplicar la transformada al bloque actual.

El aparato de decodificación deriva la muestra residual basándose en si aplicar la transformada y la información residual (E1220).

Como una realización, el aparato de decodificación recibe la información de imagen que incluye la información residual. Como se ha descrito anteriormente, la información residual puede incluir información de valor de los coeficientes de transformada cuantificados, información de posición, técnica de transformada, núcleo de transformada e información de parámetro de cuantificación. El aparato de decodificación puede derivar los coeficientes de transformada cuantificados para el bloque actual basándose en la información de coeficientes

de transformada cuantificados incluidos en la información residual, y puede derivar los coeficientes de transformada basándose en los coeficientes de transformada cuantificados. Además, el aparato de decodificación puede derivar las muestras residuales en base a los coeficientes de transformada.

5 Por ejemplo, si el valor de la información habilitada para el salto de transformada es 1 (es decir, para la información habilitada para el salto de transformada que representa que el salto de transformada está habilitado), el aparato de decodificación obtiene la información sobre si aplicar la transformada al bloque actual (indicador de salto de transformada) a partir de la sintaxis de la unidad de transformada. En este caso, el aparato de decodificación deriva las muestras residuales basándose en la información de indicador de salto de transformada. Por ejemplo, el modo de salto de transformada se aplica al bloque actual cuyo valor del indicador de salto de transformada es 1 y, en este caso, el aparato de decodificación deriva los coeficientes de transformada como las muestras residuales del bloque actual. Además, la transformada se aplica al bloque actual cuyo valor del indicador de salto de transformada es 0 y, en este caso, el aparato de decodificación deriva las muestras residuales del bloque actual a través de la transformada inversa de los coeficientes de transformada.

15 Además, con respecto al bloque actual cuyo valor del indicador de salto de transformada es 1 (es decir, modo de salto de transformada), el aparato de decodificación puede derivar el parámetro de cuantificación que se usa en el proceso de descuantificación basándose en la información de parámetro de cuantificación mínimo. Además, el aparato de decodificación puede derivar los coeficientes de transformada descuantificados llevando a cabo el proceso de descuantificación basándose en el parámetro de cuantificación, y puede derivar las muestras residuales basándose en los coeficientes de transformada descuantificados.

25 En este caso, como se ha descrito anteriormente, la información de parámetro de cuantificación mínimo es información relacionada con el parámetro de cuantificación mínimo permitido para el modo de salto de transformada, y puede incluirse en la información de imagen (p. ej., SPS) basándose en al menos una de la información habilitada de salto de transformada (p. ej., `sps_transform_skip_enabled_flag`) y/o la información habilitada de codificación de paleta (p. ej., `sps_palette_enabled_flag`). Por ejemplo, en base a la condición de que el valor de la información habilitada para el salto de transformada (p. ej., `sps_transform_skip_enabled_flag`) sea 1, o el valor de la información habilitada para la codificación de paleta (p. ej., `sps_palette_enabled_flag`) sea 1, la información de parámetro de cuantificación mínimo (p. ej., `min_qp_prime_ts_minus4`) se incluye en el SPS. Es decir, solo en el caso de que se cumpla la condición anterior, la información de parámetro de cuantificación mínimo (p. ej., `min_qp_prime_ts_minus4`) se analiza/señaliza.

35 El aparato de decodificación genera las muestras reconstruidas basándose en las muestras residuales (E1230).

40 Como una realización, el aparato de decodificación puede determinar si llevar a cabo la interpretación o la intrapredicción para el bloque actual basándose en la información de predicción (p. ej., información de modo de predicción) incluida en la información de imagen, y puede derivar las muestras de predicción para el bloque actual llevando a cabo la predicción según la determinación. Además, el aparato de decodificación puede generar las muestras reconstruidas basándose en las muestras de predicción y las muestras residuales. En este caso, el aparato de decodificación puede usar directamente las muestras de predicción como muestras reconstruidas según el modo de predicción, o puede generar las muestras reconstruidas añadiendo las muestras residuales a las muestras de predicción. Además, el aparato de decodificación puede derivar un bloque reconstruido o una imagen reconstruida basándose en las muestras reconstruidas. A continuación, según sea necesario, el aparato de decodificación puede aplicar el procedimiento de filtrado en bucle como, por ejemplo, el procedimiento de filtrado de desbloqueo y/o SAO, a la imagen reconstruida para mejorar la calidad de imagen subjetiva/objetiva como se describió anteriormente.

50 En las realizaciones descritas anteriormente, los métodos se explican basándose en diagramas de flujo por medio de una serie de etapas o bloques, pero el presente documento no se limita al orden de etapas, y puede llevarse a cabo una determinada etapa en orden o etapa diferente de la descrita anteriormente, o simultáneamente con otras etapas. Además, una persona con experiencia ordinaria en la técnica puede entender que las etapas mostradas en un diagrama de flujo no son excluyentes, y que se puede incorporar otra etapa o se pueden eliminar una o más etapas del diagrama de flujo sin afectar al alcance del presente documento.

55 El método según el presente documento puede implementarse en forma de software, y el aparato de codificación y/o el aparato de decodificación según el presente documento pueden incluirse en un aparato que lleva a cabo el procesamiento de imágenes como, por ejemplo, TV, un ordenador, un teléfono inteligente, un decodificador y un aparato de visualización.

60 Cuando las realizaciones del presente documento se implementan mediante software, el método descrito anteriormente puede implementarse mediante un módulo (proceso o función) que lleva a cabo la función descrita anteriormente. El módulo puede almacenarse en una memoria y ejecutarse por un procesador. La memoria puede instalarse dentro o fuera del procesador y puede conectarse al procesador a través de diversos medios conocidos. El procesador puede incluir un circuito integrado para aplicaciones específicas (ASIC, por sus siglas en inglés), otros conjuntos de chips, un circuito lógico y/o un dispositivo de procesamiento de datos.

La memoria puede incluir una memoria de solo lectura (ROM, por sus siglas en inglés), una memoria de acceso aleatorio (RAM, por sus siglas en inglés), una memoria flash, una tarjeta de memoria, un medio de almacenamiento y/u otro dispositivo de almacenamiento. En otras palabras, las realizaciones según el presente documento pueden implementarse y ejecutarse en un procesador, un microprocesador, un controlador o un chip. Por ejemplo, las unidades funcionales ilustradas en las figuras respectivas pueden implementarse y ejecutarse en un ordenador, un procesador, un microprocesador, un controlador o un chip. En este caso, la información sobre la implementación (por ejemplo, información sobre instrucciones) o algoritmos se pueden almacenar en un medio de almacenamiento digital.

Además, el aparato de decodificación y el aparato de codificación a los que se aplica el presente documento pueden incluirse en un transceptor 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 de vídeo y un dispositivo de comunicación en tiempo real como, por ejemplo, comunicación de vídeo, un dispositivo de transmisión continua móvil, un medio de almacenamiento, una videocámara, un proveedor de servicios de vídeo bajo demanda (VoD, por sus siglas en inglés), un dispositivo de vídeo *over the top* (OTT, por sus siglas en inglés), un proveedor de servicios de transmisión continua de Internet, un dispositivo de vídeo 3D, un dispositivo de realidad virtual (VR), un dispositivo de realidad aumentada (AR), un dispositivo de vídeo de teléfono de imagen, un terminal de vehículo (por ejemplo, un terminal de vehículo (incluido un vehículo autónomo), un terminal de avión o un terminal de barco) y un dispositivo de vídeo médico; y pueden usarse para procesar una señal o datos de imagen. Por ejemplo, el dispositivo de vídeo OTT puede incluir una consola de juegos, un reproductor Blu-ray, un TV conectada a Internet, un sistema de cine en casa, un teléfono inteligente, un ordenador de tableta y un grabador de vídeo digital (DVR, por sus siglas en inglés).

Además, el método de procesamiento al que se aplica el presente documento puede producirse en forma de un programa ejecutado por un ordenador y puede almacenarse en un medio de grabación legible por ordenador. Los datos multimedia que tienen una estructura de datos según el presente documento también pueden almacenarse en el medio de grabación legible por ordenador. El medio de grabación legible por ordenador incluye todo tipo de dispositivos de almacenamiento y dispositivos de almacenamiento distribuido en los que se almacenan datos legibles por ordenador. El medio de grabación legible por ordenador puede incluir, por ejemplo, un disco Blu-ray (BD), un bus universal en serie (USB, por sus siglas en inglés), una ROM, una PROM, una EPROM, una EEPROM, una RAM, un CD-ROM, una cinta magnética, un disquete y un dispositivo de almacenamiento de datos ópticos. El medio de grabación legible por ordenador también incluye medios incorporados en forma de una onda portadora (por ejemplo, transmisión a través de Internet). Además, un flujo de bits generado por el método de codificación puede almacenarse en el medio de grabación legible por ordenador o transmitirse a través de una red de comunicación cableada o inalámbrica.

Además, la(s) realización(es) del presente documento puede(n) realizarse como un producto de programa informático basado en un código de programa, y el código de programa puede ejecutarse en un ordenador según la(s) realización(es) del presente documento. El código de programa puede almacenarse en un soporte legible por ordenador.

La FIG. 14 representa un ejemplo de un sistema de transmisión continua de contenidos al que puede aplicarse la realización del presente documento.

Con referencia a la FIG. 14, el sistema de transmisión continua de contenidos al que se aplican las realizaciones del presente documento puede incluir generalmente un servidor de codificación, un servidor de transmisión continua, un servidor web, un almacenamiento multimedia, un dispositivo de usuario y un dispositivo de entrada multimedia.

El servidor de codificación funciona para comprimir a datos digitales los contenidos introducidos desde los dispositivos de entrada multimedia como, por ejemplo, el teléfono inteligente, la cámara, la videocámara y similares, para generar un flujo de bits, y para transmitirlo al servidor de transmisión continua. Como otro ejemplo, en un caso en el que el dispositivo de entrada multimedia como, por ejemplo, el teléfono inteligente, la cámara, la videocámara o similares, genera directamente un flujo de bits, el servidor de codificación puede omitirse.

El flujo de bits puede generarse mediante un método de codificación o un método de generación de flujo de bits al que se aplican las realizaciones del presente documento. Y el servidor de transmisión continua puede almacenar temporalmente el flujo de bits en un proceso de transmisión o recepción del flujo de bits.

El servidor de transmisión continua transmite datos multimedia al equipo de usuario basándose en una solicitud del usuario a través del servidor web, que funciona como un instrumento que informa a un usuario sobre qué servicio hay. Cuando el usuario solicita un servicio que desea el usuario, el servidor web transfiere la solicitud al servidor de transmisión continua, y el servidor de transmisión continua transmite datos multimedia al usuario. A este respecto, el sistema de transmisión continua de contenidos puede incluir un servidor de control separado y, en este caso, el servidor de control funciona para controlar comandos/respuestas entre equipos respectivos en el sistema de transmisión continua de contenidos.

El servidor de transmisión continua puede recibir contenidos del almacenamiento de medios y/o el servidor de codificación. Por ejemplo, en un caso en el que los contenidos se reciben del servidor de codificación, los contenidos pueden recibirse en tiempo real. En este caso, el servidor de transmisión continua puede almacenar el flujo de bits durante un período predeterminado para proveer el servicio de transmisión continua sin problemas.

5

Por ejemplo, el equipo de usuario puede incluir un teléfono móvil, un teléfono inteligente, un ordenador portátil, un terminal de radiodifusión digital, un asistente digital personal (PDA, por sus siglas en inglés), un reproductor multimedia portátil (PMP, por sus siglas en inglés), una navegación, un PC de pizarra, un PC de tableta, un *ultrabook*, un dispositivo ponible (p. ej., un terminal tipo reloj (reloj inteligente), un terminal tipo gafas (gafas inteligentes), una pantalla montada en la cabeza (HMD, por sus siglas en inglés)), un TV digital, un ordenador de sobremesa, una señalización digital o similares.

10

Cada uno de los servidores en el sistema de transmisión continua de contenidos puede ser operado como un servidor distribuido y, en este caso, los datos recibidos por cada servidor pueden ser procesados de manera distribuida.

15

REIVINDICACIONES

1. Un método de decodificación de imágenes para codificación versátil de vídeo, VVC, llevado a cabo por un aparato de decodificación, comprendiendo el método:

- 5 obtener información de imagen que incluye información residual e información habilitada de salto de transformada de un flujo de bits;
- 10 obtener una información de indicador de salto de transformada basándose en un valor de la información habilitada de salto de transformada, en donde la información habilitada de salto de transformada está relacionada con si el salto de transformada está habilitado, e i) el valor de la información habilitada de salto de transformada igual a 1 representa que el salto de transformada está habilitado y la información de indicador de salto de transformada está presente en una sintaxis de unidad de transformada, y ii) el valor de la información habilitada de salto de transformada igual a 0 representa que el salto de transformada no está habilitado y la información de indicador de salto de transformada no está presente en la sintaxis de unidad de transformada;
- 15 determinar si se aplica una transformada a un bloque actual basándose en la información de indicador de salto de transformada;
- 20 derivar una muestra residual en base a si se aplica la transformada y la información residual; y
- generar una muestra reconstruida a partir de la muestra residual,
- 25 en donde la información de imagen incluye información habilitada de codificación de paleta para representar si un modo de paleta está habilitado, y en donde i) un valor de la información habilitada de codificación de paleta igual a 1 representa que el modo de paleta está habilitado, y ii) el valor de la información habilitada de codificación de paleta igual a 0 representa que el modo de paleta no está habilitado,
- 30 en donde, basándose en una condición de que el valor de la información habilitada de salto de transformada sea igual a 1 y una condición de que el valor de la información habilitada de codificación de paleta sea igual a 1, la información de imagen incluye además información de parámetro de cuantificación mínimo relacionada con un parámetro de cuantificación mínimo permitido para un modo de salto de transformada, y en donde, basándose en una condición de que tanto el valor de la información habilitada de salto de transformada como el valor de la información habilitada de codificación de paleta sean iguales a 0, la información de parámetro de cuantificación mínimo no está incluida en la información de imagen,
- 35 en donde, en base a un caso donde el salto de transformada se aplica al bloque actual, un parámetro de cuantificación para el bloque actual se deriva en base a la información del parámetro de cuantificación mínimo, y en donde, en base a un caso donde el modo de paleta se aplica al bloque actual, el parámetro de cuantificación para el bloque actual se deriva en base a la información de parámetro de cuantificación mínimo,
- 40 en donde, en base al caso donde el salto de transformada se aplica al bloque actual, la muestra residual se deriva en base al parámetro de cuantificación, y en base al caso donde el modo de paleta se aplica al bloque actual, un valor de escape se deriva en base al parámetro de cuantificación.
- 45

2. Un método de codificación de imágenes para codificación de vídeo versátil, VVC, llevado a cabo por un aparato de codificación, comprendiendo el método:

- 50 determinar si aplicar una transformada a un bloque actual basándose en información habilitada para salto de transformada;
- generar una información de indicador de salto de transformada basándose en si se aplica el salto de transformada;
- 55 generar información residual en el bloque actual basándose en si se aplica la transformada; y
- codificar la información de imagen que incluye la información habilitada para el salto de transformada y la información residual,
- 60 en donde la información habilitada para el salto de transformada está relacionada con si el salto de transformada está habilitado, e i) un valor de la información habilitada para el salto de transformada igual a 1 representa que el salto de transformada está habilitado y la información de indicador de salto de transformada está presente en una sintaxis de unidad de transformada, y ii) el valor de la información habilitada para el salto de transformada igual a 0 representa que el salto de transformada no está habilitado y la información de indicador de salto de transformada no está presente en la sintaxis de unidad de transformada,
- 65

en donde la información de imagen incluye información habilitada de codificación de paleta para representar si un modo de paleta está habilitado, y en donde i) un valor de la información habilitada de codificación de paleta igual a 1 representa que el modo de paleta está habilitado, y ii) el valor de la información habilitada de codificación de paleta igual a 0 representa que el modo de paleta no está habilitado,

5

en donde, basándose en una condición de que el valor de la información habilitada de salto de transformada sea igual a 1 y una condición de que el valor de la información habilitada de codificación de paleta sea igual a 1, la información de imagen incluye además información de parámetro de cuantificación mínimo relacionada con un parámetro de cuantificación mínimo permitido para un modo de salto de transformada, y en donde, basándose en una condición de que tanto el valor de la información habilitada de salto de transformada como el valor de la información habilitada de codificación de paleta sean iguales a 0, la información de parámetro de cuantificación mínimo no está incluida en la información de imagen,

10

en donde, basándose en un caso donde el salto de transformada se aplica al bloque actual, un parámetro de cuantificación para el bloque actual se determina basándose en la información de parámetro de cuantificación mínimo, y en donde, basándose en un caso donde el modo de paleta se aplica al bloque actual, el parámetro de cuantificación para el bloque actual se determina basándose en la información de parámetro de cuantificación mínimo.

15

3. Un medio de almacenamiento legible por ordenador no transitorio que almacena un flujo de bits para codificación de vídeo versátil, VVC, generado por un método, comprendiendo el método:

20

llevar a cabo la determinación de si se aplica una transformada a un bloque actual basándose en información habilitada para el salto de transformada, generar una información de indicador de salto de transformada basándose en si se aplica la transformada, generar información residual en el bloque actual basándose en si se aplica el salto de transformada, y generar el flujo de bits codificando información de imagen que incluye la información habilitada para el salto de transformada y la información residual,

25

en donde la información habilitada para el salto de transformada está relacionada con si el salto de transformada está habilitado, e i) un valor de la información habilitada para el salto de transformada igual a 1 representa que el salto de transformada está habilitado y la información de indicador de salto de transformada está presente en una sintaxis de unidad de transformada, y ii) el valor de la información habilitada para el salto de transformada igual a 0 representa que el salto de transformada no está habilitado y la información de indicador de salto de transformada no está presente en la sintaxis de unidad de transformada,

30

35

en donde la información de imagen incluye información habilitada de codificación de paleta para representar si un modo de paleta está habilitado, y en donde i) un valor de la información habilitada de codificación de paleta igual a 1 representa que el modo de paleta está habilitado, y ii) el valor de la información habilitada de codificación de paleta igual a 0 representa que el modo de paleta no está habilitado,

40

en donde, basándose en una condición de que el valor de la información habilitada de salto de transformada sea igual a 1 y una condición de que el valor de la información habilitada de codificación de paleta sea igual a 1, la información de imagen incluye además información de parámetro de cuantificación mínimo relacionada con un parámetro de cuantificación mínimo permitido para un modo de salto de transformada, y en donde, basándose en una condición de que tanto el valor de la información habilitada de salto de transformada como el valor de la información habilitada de codificación de paleta sean iguales a 0, la información de parámetro de cuantificación mínimo no está incluida en la información de imagen,

45

en donde, basándose en un caso donde el salto de transformada se aplica al bloque actual, un parámetro de cuantificación para el bloque actual se determina basándose en la información de parámetro de cuantificación mínimo, y en donde, basándose en un caso donde el modo de paleta se aplica al bloque actual, el parámetro de cuantificación para el bloque actual se determina basándose en la información de parámetro de cuantificación mínimo.

50

4. Un método de transmisión de datos para imágenes, para codificación versátil de vídeo, VVC, comprendiendo el método de transmisión:

55

generar un flujo de bits llevando a cabo el método de codificación de imágenes de la reivindicación 2; y

60

transmitir los datos que comprenden el flujo de bits.

DIBUJOS

FIG. 1

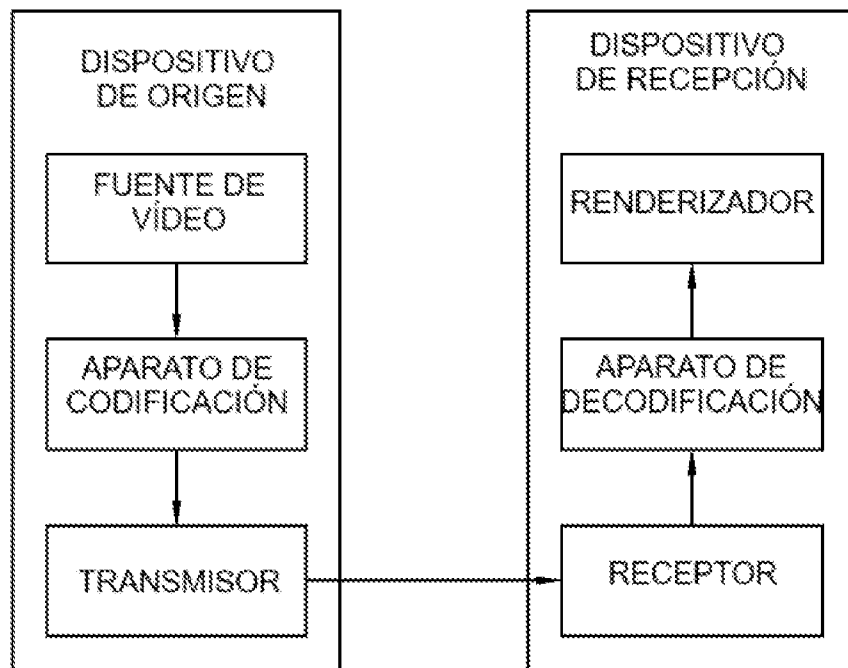


FIG. 2

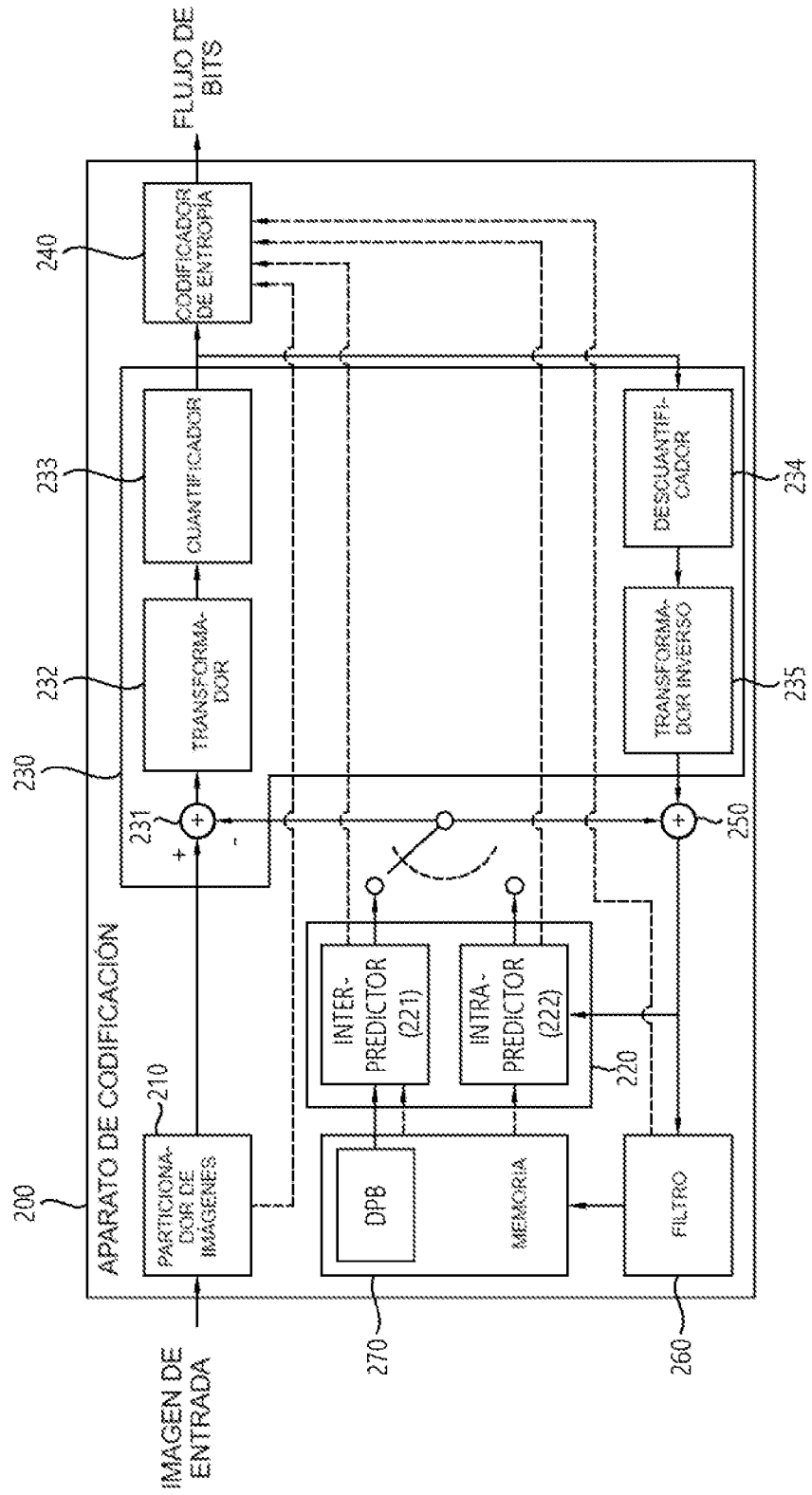


FIG. 3

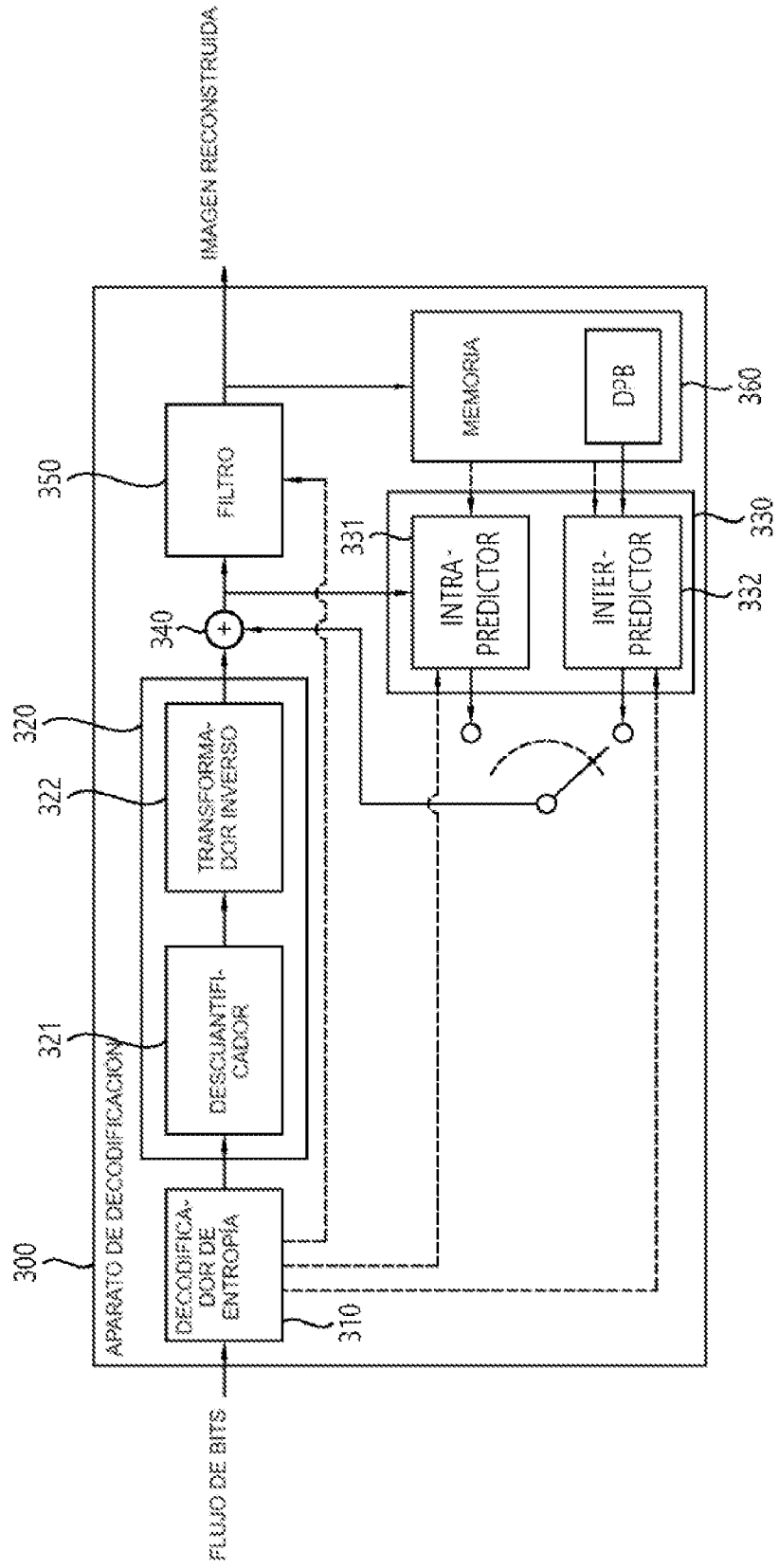


FIG. 4

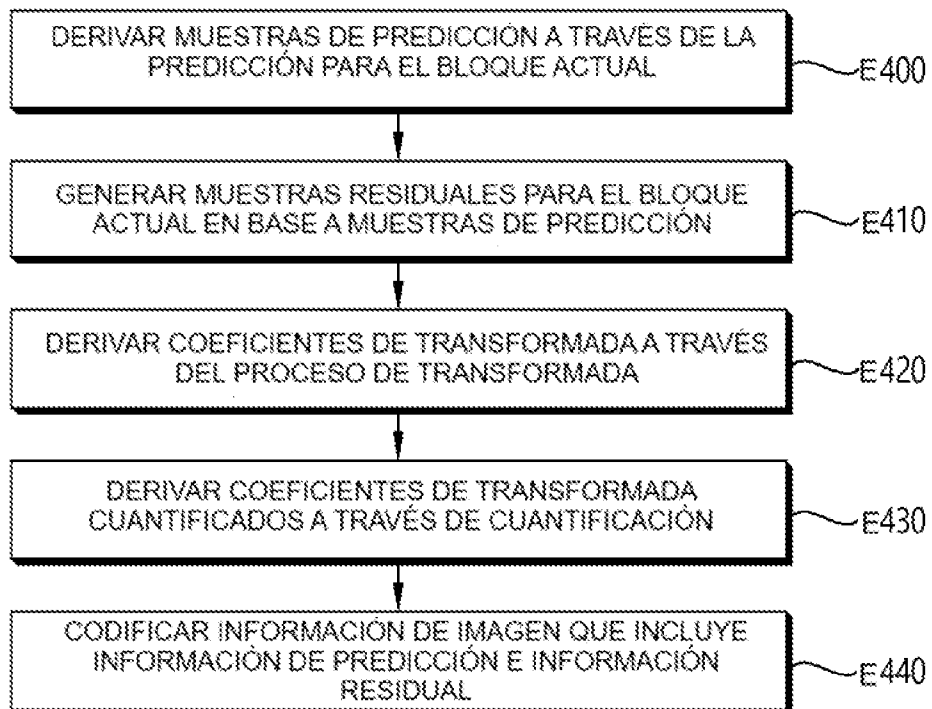


FIG. 5

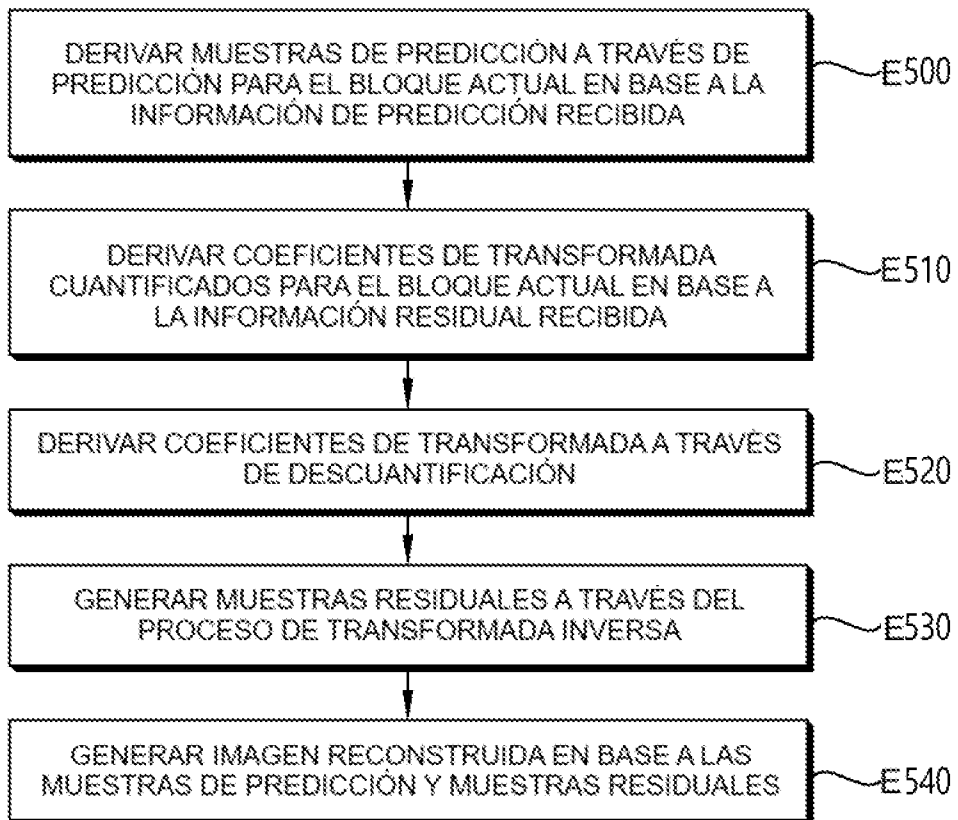


FIG. 6



FIG. 7

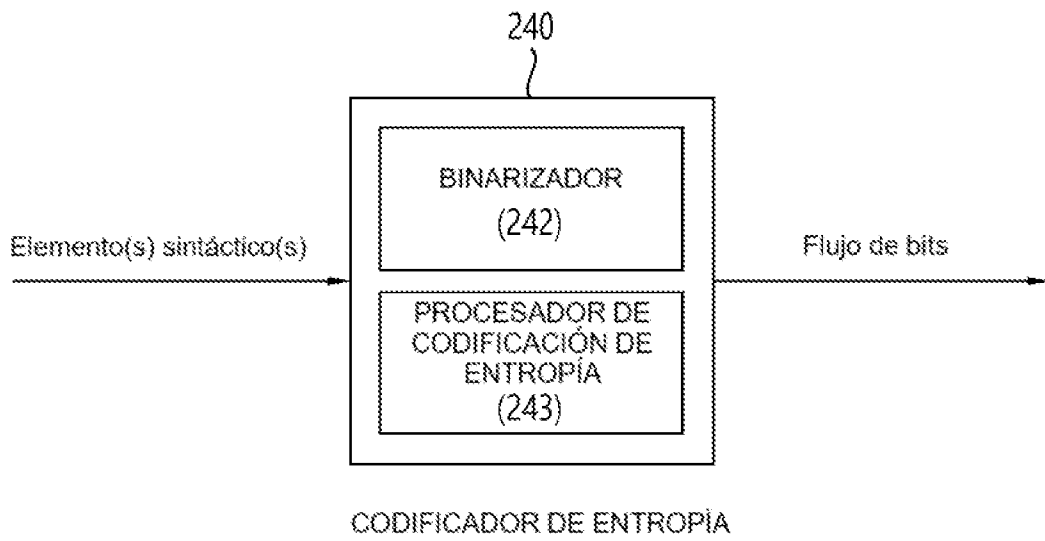


FIG. 8

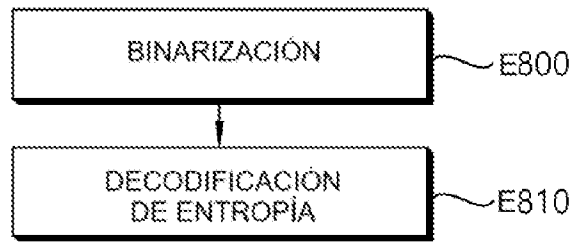


FIG. 9

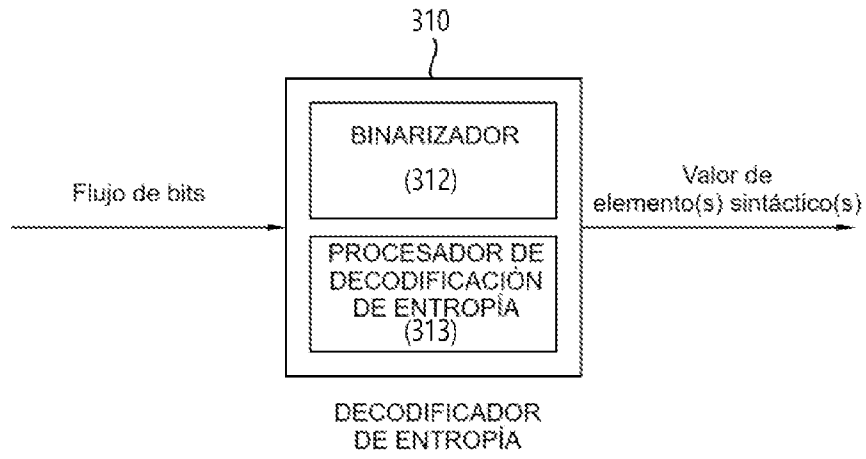


FIG. 10

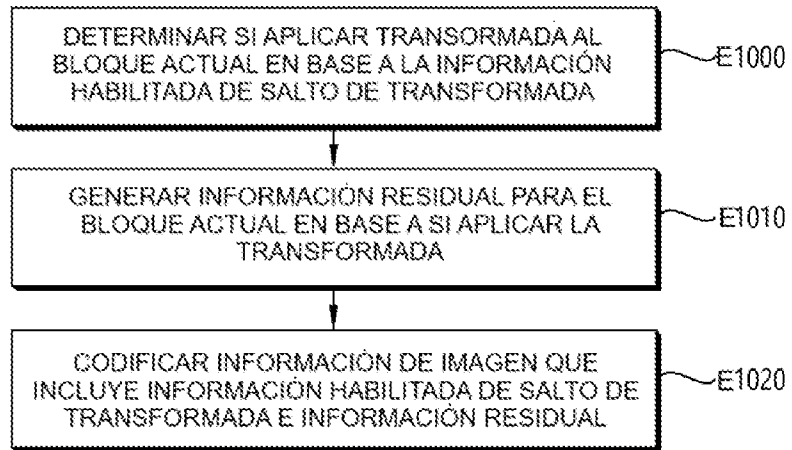


FIG. 11

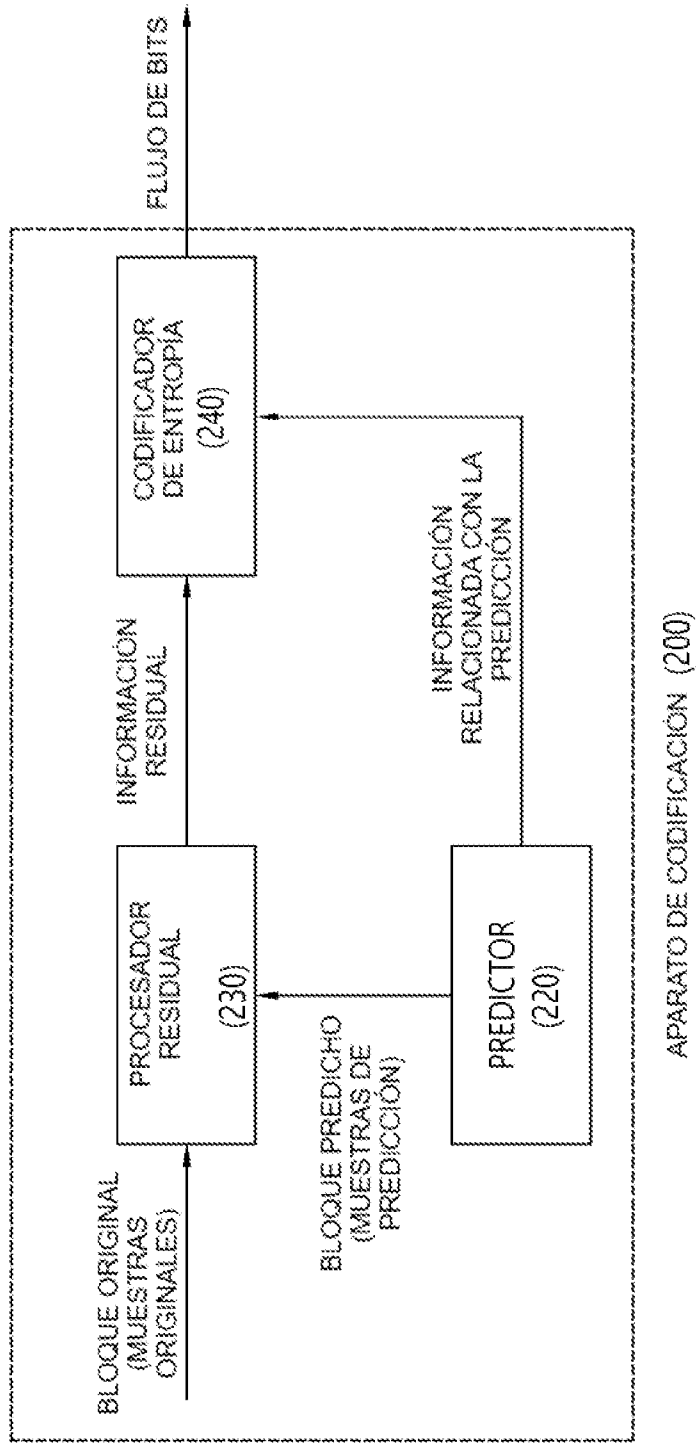


FIG. 12

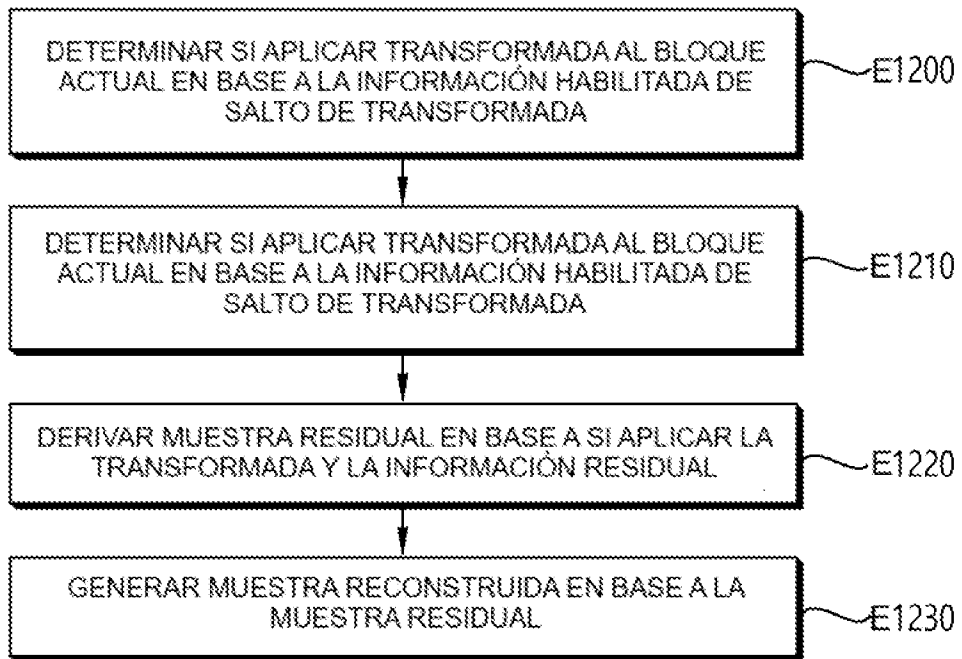


FIG. 13

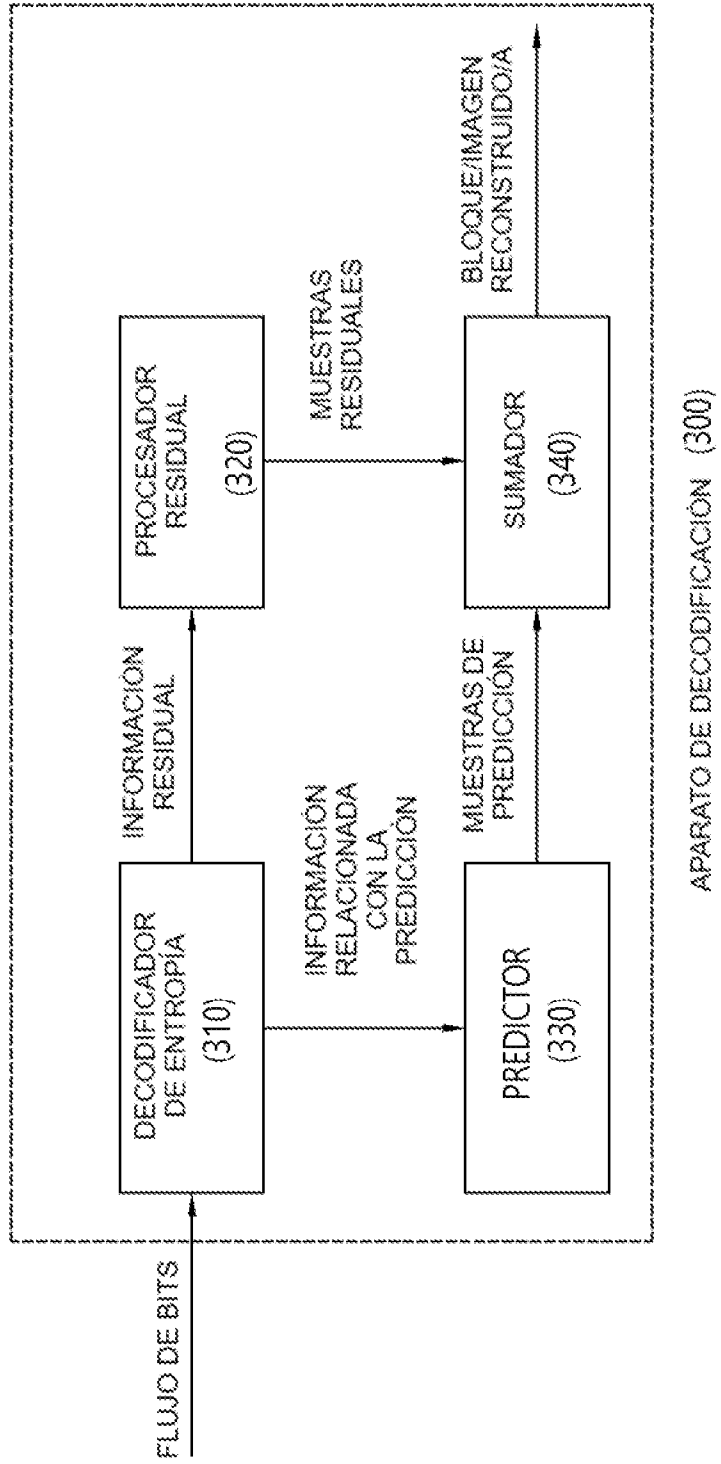


FIG. 14

