

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 975 180**

51 Int. Cl.:

H04N 19/176 (2014.01)
H04N 19/70 (2014.01)
H04N 19/134 (2014.01)
H04N 19/126 (2014.01)
H04N 19/14 (2014.01)
H04N 19/186 (2014.01)
H04N 19/157 (2014.01)
H04N 19/18 (2014.01)
H04N 19/184 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 86 Fecha de presentación y número de la solicitud internacional: **13.03.2015** **PCT/US2015/020535**
87 Fecha y número de publicación internacional: **17.09.2015** **WO15138957**
96 Fecha de presentación y número de la solicitud europea: **13.03.2015** **E 15713605 (2)**
97 Fecha y número de publicación de la concesión europea: **28.02.2024** **EP 3117610**

54 Título: **Modificación de las profundidades de bits en la codificación de transformación de espacio de color**

30 Prioridad:

14.03.2014 US 201461953573 P
18.04.2014 US 201461981645 P
10.10.2014 US 201462062637 P
12.03.2015 US 201514656526

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
03.07.2024

73 Titular/es:

QUALCOMM INCORPORATED (100.0%)
Attn: International IP Administration 5775
Morehouse Drive
San Diego, California 92121-1714, US

72 Inventor/es:

ZHANG, LI;
CHEN, JIANLE;
KARCZEWICZ, MARTA;
SOLE ROJALS, JOEL y
KIM, WOO-SHIK

74 Agente/Representante:

ISERN JARA, Jorge

ES 2 975 180 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

Modificación de las profundidades de bits en la codificación de transformación de espacio de color

5 Esta solicitud reivindica el beneficio de Solicitud Provisional de EE. UU. No. 61/953,573, presentada el 14 de marzo de 2014, Solicitud Provisional de EE. UU. No. 61/981,645, presentada el 18 de abril de 2014, y Solicitud Provisional de EE. UU. No. 62/062,637, presentada el 10 de octubre de 2014.

Campo técnico

10 Esta divulgación se relaciona con la codificación de vídeo y, más específicamente, la codificación de vídeo usando conversión de espacio de color.

Antecedentes

15 Las capacidades de vídeo digital se pueden incorporar a una amplia gama de dispositivos, incluidos televisores digitales, sistemas de transmisión directa digital, sistemas de transmisión inalámbricos, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, tabletas, lectores de libros electrónicos, cámaras digitales, dispositivos de grabación, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos celulares o radiotelefonos satelitales, los llamados "teléfonos inteligentes", dispositivos de vídeo teleconferencia, dispositivos de transmisión de vídeo y similares. Los dispositivos de vídeo digital implementan técnicas de codificación de vídeo, tal como las descritas en los estándares definidos por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Parte 10, Codificación de vídeo avanzada (AVC), el estándar de codificación de vídeo de alta eficiencia (HEVC) actualmente en desarrollo, y extensiones de dichos estándares. Los dispositivos de vídeo pueden transmitir, recibir, codificar, decodificar y/o almacenar información de vídeo digital de manera más eficiente implementando dichas técnicas de codificación de vídeo.

30 Las técnicas de codificación de vídeo incluyen predicción espacial (imagen intra) y/o predicción temporal (entre imágenes) para reducir o eliminar la redundancia inherente a las secuencias de vídeo. Para la codificación de vídeo basada en bloques, una porción de vídeo (por ejemplo, una trama de vídeo o una porción de una trama de vídeo) puede dividirse en bloques de vídeo, que también pueden denominarse bloques de árbol, unidades de codificación (CU) y/o nodos de codificación. Los bloques de vídeo en una porción codificada intra (I) de una imagen se codifican mediante predicción espacial con respecto a muestras de referencia en bloques vecinos en la misma imagen. Los bloques de vídeo en una porción codificada inter (P o B) de una imagen pueden usar predicción espacial con respecto a muestras de referencia en bloques vecinos en la misma imagen o predicción temporal con respecto a muestras de referencia en otras imágenes de referencia. Las imágenes pueden denominarse tramas y las imágenes de referencia pueden denominarse tramas de referencia.

40 La predicción espacial o temporal da como resultado un bloque predictivo para un bloque a codificar. Los datos residuales representan diferencias de píxeles entre el bloque original que se va a codificar y el bloque predictivo. Un bloque codificado inter se codifica de acuerdo con un vector de movimiento que apunta a un bloque de muestras de referencia que forman el bloque predictivo, y los datos residuales indican la diferencia entre el bloque codificado y el bloque predictivo. Un bloque codificado intra se codifica de acuerdo con un modo de codificación intra y los datos residuales. Para una mayor compresión, los datos residuales pueden transformarse desde el dominio de píxeles a un dominio de transformación, dando como resultado coeficientes de transformación residuales, que luego pueden cuantificarse. Los coeficientes de transformación cuantificados, inicialmente dispuestos en una matriz bidimensional, se pueden explorar para producir un vector unidimensional de coeficientes de transformación, y se puede aplicar codificación de entropía para lograr aún más compresión.

50 El siguiente documento: Zhang et al: "SCCE5 Test 3.2.1: In loop color-space transform", 18. JCT-VC Meeting; 30-6-2014 - 9-7-2014; Sapporo; (Joint Collaborative Team on Video Coding of ISO/IEC JTC1/SC29/WG11 and ITU-T SG.16); no. JCTVC-R0147, 20 de junio de 2014 describe una transformación del espacio de color realizada en el dominio residual. La transformación del espacio de color se puede utilizar para bloques codificados en modo intra. Las transformaciones de espacio de color directa e inversa para codificación con pérdida utilizan las matrices de transformación YCoCg. Para codificación sin pérdidas, se utiliza la transformación YCoCg-R. El espacio de color original (C_0 , C_1 , C_2) puede corresponder a (R, G, B) o (Y, U, V). La transformación de color directa en la codificación con pérdida no está normalizada, lo que da como resultado una disminución de la señal cuando se aplica la transformación. Los QP delta de (-5, -3, -5) se utilizan para compensar dicha disminución para los tres componentes de color, respectivamente. Es decir, cuando se aplica la transformación del espacio de color, el parámetro de cuantificación se establece igual a (QP - 5, QP - 3, QP - 5) para los tres componentes, respectivamente, donde QP es el valor QP "normal" para el CU. El siguiente documento: Rabbani M et al: "Una descripción general del estándar de compresión de imágenes fijas JPEG 2000", Signal Processing Image Communication, Elsevier Science Publishers, Amsterdam, NL, vol. 17, núm. 1, enero de 2002, páginas 3-48, ISSN: 0923-5965, DOI: 10.1016/S0923-5965(01)00024-8 describe, para JPEG 2000, una transformación de color reversible que puede usarse para codificación tanto sin pérdida como con pérdida.

Resumen

En general, esta divulgación describe técnicas para codificar bloques de vídeo utilizando un proceso de conversión de espacio de color. En particular, la invención está dirigida a métodos y dispositivos para codificación y decodificación de vídeo como se define en las reivindicaciones independientes 1, 4, 8 y 12, así como a un correspondiente medio de almacenamiento legible por ordenador que almacena instrucciones del procesador como se define en la reivindicación 7.

Los detalles de uno o más ejemplos de la divulgación se exponen en los dibujos adjuntos y en la descripción siguiente. Otras características, objetos y ventajas de la divulgación resultarán evidentes a partir de la descripción, los dibujos y las reivindicaciones.

Breve descripción de los dibujos

La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación y decodificación de vídeo de ejemplo que puede utilizar las técnicas descritas en esta divulgación.

La FIG. 2 es un diagrama de bloques que ilustra un codificador de vídeo de ejemplo que puede implementar las técnicas descritas en esta divulgación.

La FIG. 3 es un diagrama de bloques que ilustra un decodificador de vídeo de ejemplo que puede implementar las técnicas descritas en esta divulgación.

La FIG. 4 es un diagrama conceptual que ilustra los 35 modos de predicción HEVC de acuerdo con una o más técnicas de la presente divulgación.

La FIG. 5 es un diagrama conceptual que ilustra candidatos de vectores de movimiento vecinos espaciales para modos de fusión y predicción avanzada de vectores de movimiento (AMVP) de acuerdo con una o más técnicas de la presente divulgación.

La FIG. 6 es un diagrama conceptual que ilustra un ejemplo de copia bloque intra (BC) de acuerdo con una o más técnicas de la presente divulgación.

La FIG. 7 es un diagrama conceptual que ilustra un ejemplo de un bloque objetivo y una muestra de referencia para un bloque intra 8x8, de acuerdo con una o más técnicas de la presente divulgación.

La FIG. 8 es un diagrama de flujo que ilustra una técnica de codificación de acuerdo con una o más técnicas de la presente divulgación.

La FIG. 9 es un diagrama de flujo que ilustra una técnica de decodificación de acuerdo con una o más técnicas de la presente divulgación.

Descripción detallada

En algunos ejemplos, esta divulgación está relacionada con la codificación de contenido de pantalla, en la que se usa el formato de muestreo de alto croma 4:4:4. En algunos ejemplos, esta divulgación también es aplicable para extensiones de intervalo (RCE), incluido el soporte de una profundidad de bits posiblemente alta (más de 8 bits), formato de muestreo de croma alto 4:4:4. En algunos ejemplos, esta divulgación también es aplicable a otros formatos de color, tal como el formato de muestreo de croma es 4:2:2. Más específicamente, en esta divulgación se describen muchas técnicas diferentes relacionadas con la conversión del espacio de color.

Los estándares de codificación de vídeo incluyen ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 o ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual e ITU-T H.264 (también conocido como ISO/IEC MPEG-4 AVC), incluyendo sus extensiones de Codificación de Vídeo Escalable (SVC) y Codificación de Vídeo Multivista (MVC).

El diseño de un nuevo estándar de codificación de vídeo, es decir, Codificación de Vídeo de Alta Eficiencia (HEVC), ha sido finalizado por el Equipo Conjunto de Colaboración sobre Codificación de Vídeo (JCT-VC) del Grupo de Expertos en Codificación de Vídeo (VCEG) del ITU-T y el Grupo de Expertos en Películas de ISO/IEC (MPEG).

En HEVC, la unidad de codificación más grande de un segmento se denomina bloque de árbol de codificación (CTB). Un CTB contiene un árbol cuádruple cuyos nodos son unidades de codificación. El tamaño de un CTB puede oscilar entre 16x16 y 64x64 en el perfil principal HEVC, aunque se pueden admitir tamaños de CTB de 8x8. Una unidad de codificación (CU) podría tener el mismo tamaño que una CTB y tan pequeña como 8x8. Cada CU está codificada con un modo. Cuando una CU está codificada inter, la CU puede dividirse además en dos unidades de predicción (PU) o convertirse en una única PU cuando no se aplican más particiones. Cuando hay dos PU en una CU, cada PU puede ser rectángulos de tamaño medio o dos rectángulos con un tamaño igual a 1/4 o 3/4 del tamaño de la CU.

Cuando la CU está codificada inter, está presente un conjunto de información de movimiento para cada PU. Además, cada PU está codificada con un modo de predicción inter único para derivar el conjunto de información de movimiento. En HEVC, los tamaños de PU más pequeños son 8x4 y 4x8.

HEVC especifica cuatro unidades de transformación (TU) con tamaños de 4x4, 8x8, 16x16 y 32x32 para codificar el residual de predicción. Un CTB puede dividirse de forma recursiva en 4 o más TU. Las TU utilizan funciones basadas en números enteros que son similares a las funciones de transformación de coseno discreta (DCT). Además, los

bloques de transformación luma 4x4 que pertenecen a una región codificada intra se transforman utilizando una transformación entera que se deriva de una función de transformación sinusoidal discreta (DST). Croma usa los mismos tamaños de TU que luma.

- 5 En el estándar HEVC actual, para el componente luma de cada Unidad de Predicción (PU), se utiliza un método de predicción intra con 33 modos de predicción angular (indexados de 2 a 34), un modo DC (indexado con 1) y un modo Planar (indexado con 0), como se describe a continuación con respecto a la FIG. 4.

10 Además de los 35 modos intra anteriores, HEVC también emplea un modo más, denominado 'I-PCM'. En el modo I-PCM, la predicción, la transformación, la cuantificación y la codificación de entropía se omiten mientras que las muestras de predicción se codifican mediante un número predefinido de bits. El objetivo principal del modo I-PCM es manejar la situación en la que otros modos no pueden codificar eficientemente la señal.

- 15 En el estándar HEVC actual, hay dos modos de predicción inter disponibles. Uno de esos modos es el modo de fusión (el salto se considera un caso especial de fusión), y el segundo modo es el modo de predicción avanzada de vectores de movimiento (AMVP) para una unidad de predicción (PU).

20 Tanto en el modo AMVP como en el modo de combinación, se mantiene una lista de candidatos de vector de movimiento (MV) para múltiples predictores de vectores de movimiento. Los vectores de movimiento, así como los índices de referencia en el modo de fusión, de la PU actual se generan tomando un candidato de la lista de candidatos de MV.

25 La lista de candidatos MV puede contener hasta 5 candidatos para el modo de fusión y sólo dos candidatos para el modo AMVP. Un candidato de fusión puede contener un conjunto de información de movimiento, por ejemplo, vectores de movimiento correspondientes tanto a listas de imágenes de referencia (lista 0 y lista 1) como a los índices de referencia. Si un candidato de fusión se identifica mediante un índice de fusión, se utilizan las imágenes de referencia para la predicción de los bloques actuales, así como se determinan los vectores de movimiento asociados. Sin embargo, en el modo AMVP para cada dirección de predicción potencial de la lista 0 o la lista 1, es necesario señalar explícitamente un índice de referencia, junto con un índice MVP para la lista de candidatos MV, ya que el candidato AMVP contiene sólo un vector de movimiento. En el modo AMVP, los vectores de movimiento predichos se pueden refinar aún más.

35 Un candidato de fusión corresponde a un conjunto completo de información de movimiento, mientras que un candidato AMVP puede contener solo un vector de movimiento para una dirección de predicción y un índice de referencia específicos. Los candidatos para ambos modos se derivan de manera similar de los mismos bloques vecinos espaciales y temporales.

40 Como se describe a continuación con respecto a la FIG. 5, los candidatos de MV espaciales se derivan de los bloques vecinos mostrados en la FIG. 5, para una PU específica (PU0). Sin embargo, los métodos que generan los candidatos a partir de los bloques difieren para los modos de fusión y AMVP.

45 En el modo de fusión, se pueden derivar hasta cuatro candidatos MV espaciales con los órdenes mostrados en la FIG. 5(a) con números. El orden es el siguiente: izquierda (0), arriba (1), arriba derecha (2), abajo izquierda (3) y arriba izquierda (4), como se muestra en la FIG. 5(a).

50 En el modo AMVP, los bloques vecinos se dividen en dos grupos: un grupo izquierdo que incluye los bloques 0 y 1, y un grupo superior que incluye los bloques 2, 3 y 4 como se muestra en la FIG. 5(b). Para cada grupo, el candidato potencial en un bloque vecino que hace referencia a la misma imagen de referencia que la indicada por el índice de referencia señalado tiene la prioridad más alta para ser elegido para formar un candidato final del grupo. Es posible que todos los bloques vecinos no contengan un vector de movimiento que apunte a la misma imagen de referencia. Por lo tanto, si no se puede encontrar dicho candidato, el primer candidato disponible se escalará para formar el candidato final, de modo que se puedan compensar las diferencias de distancia temporal.

55 Como se describe a continuación con respecto a la FIG. 6, la copia bloque intra (BC) se ha incluido en SCC. Se muestra un ejemplo de Intra BC en la FIG. 6, en el que la CU actual se predice a partir de un bloque ya decodificado de la imagen/segmento actual. El tamaño de bloque Intra BC actual puede ser tan grande como el tamaño de una CU, que oscila entre 8x8 y 64x64, aunque en algunas aplicaciones pueden aplicarse restricciones adicionales.

60 En algunos ejemplos, se puede utilizar un proceso de transformación del espacio de color en bucle para señales residuales para secuencias en formato cromático 4:4:4. Este método transforma las señales de error de predicción en formato cromático RGB/YUV en aquellas en un espacio de color subóptimo. Con este paso adicional, la correlación entre los componentes del color podría reducirse aún más. La matriz de transformación se deriva de valores de muestra de píxeles para cada unidad de codificación mediante una descomposición de valores singulares (SVD). La transformación del espacio de color se aplica al error de predicción del modo intra e inter.

65

En algunos ejemplos, cuando el proceso de transformación del espacio de color se aplica al modo inter, el residuo se convierte primero a un dominio diferente con la matriz de transformación derivada. Después de la conversión del espacio de color, los pasos de codificación convencionales, tales como DCT/DST, cuantificación y codificación de entropía, se realizan en orden.

En algunos ejemplos, cuando el proceso de transformación del espacio de color se aplica al modo intra, la predicción y el bloque actual se convierten primero a un dominio diferente con la matriz de transformación derivada, respectivamente. Después de la conversión del espacio de color, el residuo entre el bloque actual y su predictor se transforma aún más con DCT/DST, se cuantifica y se codifica con entropía.

En la operación directa, se aplica una matriz de transformación de espacio de color a los tres planos G, B y R de la siguiente manera:

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} G \\ B \\ R \end{bmatrix} = \begin{bmatrix} P \\ Q \\ S \end{bmatrix}.$$

Los valores resultantes se recortan dentro del intervalo de la especificación HEVC porque, en algunos ejemplos, los valores se amplían hasta

$\sqrt{3}$ veces. En la operación inversa, se aplica una matriz de transformación del espacio de color a los tres componentes P', Q' y R' de la siguiente manera:

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}^t \begin{bmatrix} P' \\ Q' \\ S' \end{bmatrix} = \begin{bmatrix} G' \\ B' \\ R' \end{bmatrix}.$$

Como se describe a continuación con respecto a la FIG. 7, se puede derivar una matriz de transformación a partir de los valores de muestra de referencia. Se pueden utilizar diferentes muestras de referencia para el caso intra e inter caso. Para el caso de un bloque codificado intra, se muestran un bloque objetivo y muestras de referencia en la FIG. 7. En este ejemplo, el bloque objetivo consta de muestras rayadas de 8x8 y las muestras de referencia son muestras rayadas y punteadas.

Para el caso de un bloque codificado inter, las muestras de referencia para la derivación de la matriz son las mismas que las muestras de referencia para la compensación de movimiento. Para realizar la operación de cambio, las muestras de referencia en el bloque AMP se submuestran de modo que el número de muestras se convierta en la potencia de dos. Por ejemplo, el número de muestras de referencia en un bloque de 12x16 se reduce en 2/3.

De acuerdo con las técnicas de la presente divulgación, el proceso de transformación del espacio de color se puede aplicar para cada CU. Por lo tanto, no es necesario indicar si se invoca o no el proceso de transformación. Además, tanto el codificador como el decodificador derivan la matriz de transformación con el mismo método para evitar la sobrecarga de señalización de la matriz de transformación.

De acuerdo con las técnicas de la presente divulgación, se utilizan procesos de transformación del espacio de color, tales como matrices de transformación del espacio de color. Una de esas matrices es la matriz de transformación YCbCr, que es:

Adelante:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.2126 & 0.7152 & 0.0722 \\ -0.1172 & -0.3942 & 0.5114 \\ 0.5114 & -0.4645 & -0.0469 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Inversa :

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.5397 \\ 1 & -0.1831 & -0.4577 \\ 1 & 1.8142 & 0 \end{bmatrix} \begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix}$$

Otra matriz de este tipo es la matriz de transformación YCoCg, que es:

Adelante:

$$\begin{bmatrix} Y \\ Co \\ Cg \end{bmatrix} = \begin{bmatrix} 1/4 & 1/2 & 1/4 \\ 1/2 & 0 & -1/2 \\ -1/4 & 1/2 & -1/4 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Inversa:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 1 & -1 \\ 1 & 0 & 1 \\ 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} Y \\ Co \\ Cg \end{bmatrix}$$

- 5 Otra matriz de este tipo es la matriz YCoCg-R, que es una versión revisada de la matriz YCoCg que escala los componentes Co y Cg en un factor de dos. Al utilizar una técnica de elevación, la transformación directa e inversa se podría lograr mediante las siguientes ecuaciones:

Adelante:

$$\begin{aligned} 10 \quad Co &= R - B \\ t &= B + \lfloor Co/2 \rfloor \\ Cg &= G - t \\ 15 \quad Y &= t + \lfloor Cg/2 \rfloor \end{aligned}$$

Inversa:

$$\begin{aligned} t &= Y - \lfloor Cg/2 \rfloor \\ 20 \quad G &= Cg + t \\ B &= t - \lfloor Co/2 \rfloor \\ R &= B + Co \end{aligned}$$

- 25 En las ecuaciones y matrices anteriores, las transformaciones directas se pueden realizar durante el proceso de codificación (por ejemplo, mediante un codificador de vídeo) y las transformaciones inversas se pueden realizar en el proceso de decodificación (por ejemplo, mediante un decodificador de vídeo).

- 30 En la codificación de vídeo tradicional, se supone que las imágenes tienen un tono continuo y son espacialmente suaves. Con base en estos supuestos, se han desarrollado varias herramientas (por ejemplo, transformación basada en bloques, filtrado, etc.) que han mostrado un buen rendimiento para vídeos con contenido natural. Sin embargo, en determinadas aplicaciones (por ejemplo, un escritorio remoto, pantallas de trabajo colaborativo y pantallas inalámbricas), el contenido de pantalla generado por ordenador puede ser el contenido dominante a comprimir. Este tipo de contenido tiende a tener un tono discreto y presenta líneas nítidas y límites de objetos de alto contraste. Es posible que la suposición de tono continuo y suavidad ya no se aplique y, por lo tanto, es posible que las técnicas tradicionales de codificación de vídeo no funcionen de manera eficiente.

- 40 Se puede utilizar codificación en modo paleta para superar las deficiencias anteriores. Ejemplos de técnicas de codificación de paletas se describen en la Solicitud Provisional de EE. UU. No. 61/810,649, presentada el 10 de abril de 2013. Para cada CU, se puede derivar una paleta que incluya los valores de píxeles más dominantes en la CU actual. Primero se transmiten el tamaño y los elementos de la paleta. Después de la transmisión, los píxeles de la CU se codifican de acuerdo con un determinado orden de exploración. Para cada ubicación, primero se transmite un elemento de sintaxis, tal como una bandera, `palette_flag`, para indicar si el valor del píxel está en la paleta ("modo de ejecución") o no ("modo de píxel").

- 45 En "modo de ejecución", se puede indicar el índice de la paleta, seguido del "ejecutar". La ejecución es un elemento de sintaxis que indica el número de píxeles consecutivos en un orden de exploración que tienen el mismo valor de índice de paleta que el píxel que se está codificando actualmente. Si varios píxeles en sucesión inmediata en el orden de exploración tienen el mismo valor de índice de paleta, entonces el "modo de ejecución" puede indicarse mediante

el elemento de sintaxis, tal como `palette_flag`. Se puede determinar un valor de contador, que es igual al número de píxeles posteriores al píxel actual que tienen el mismo valor de índice de paleta que el píxel actual, y la ejecución se establece igual al valor del contador. No es necesario transmitir ni `palette_flag` ni el índice de la paleta para las siguientes posiciones que están cubiertas por la "ejecución", ya que todas tienen el mismo valor de píxel. En el lado del decodificador, solo se decodificaría el primer valor del índice de paleta para el píxel actual y el resultado se duplicaría para cada píxel en la "ejecución" de píxeles indicada en el elemento de sintaxis "ejecutar".

En el "modo de píxeles" se transmite el valor de muestra de píxeles para esta posición. Si el elemento de sintaxis, como `palette_flag`, indica "modo de píxel", entonces el valor del índice de la paleta solo se determina para el píxel que se está decodificando.

Las técnicas convencionales pueden sufrir diversos problemas. Por ejemplo, se puede invocar una transformación de espacio de color que no tenga en cuenta las características de la secuencia ni la diversidad local. Por lo tanto, el rendimiento de la codificación puede ser subóptimo. En otro ejemplo, puede ser necesaria la derivación de una matriz de transformación en el decodificador, lo que aumenta significativamente la complejidad del decodificador. Además, la matriz de transformación se puede derivar utilizando los píxeles reconstruidos espacialmente o el predictor de PU codificada inter. Sin embargo, la eficiencia de la matriz de transformación puede reducirse cuando el tamaño de la PU es relativamente pequeño, la predicción no es muy precisa o los píxeles vecinos no están disponibles. Las técnicas de esta divulgación pueden superar uno o más de estos problemas.

Un codificador de vídeo, tal como un codificador 20 de vídeo o un decodificador 30 de vídeo, puede ejecutar las técnicas descritas en esta divulgación. En general, esta divulgación describe técnicas para codificar bloques de vídeo utilizando un proceso de conversión de espacio de color. Un codificador de vídeo, tal como un codificador 20 de vídeo o un decodificador 30 de vídeo, puede determinar una profundidad de bits de un componente luma de los datos de vídeo y una profundidad de bits de un componente croma de los datos de vídeo. En respuesta a que la profundidad de bits del componente luma sea diferente de la profundidad de bits del componente croma, el codificador de vídeo puede modificar una o ambas de la profundidad de bits del componente luma y la profundidad de bits del componente cromático de modo que las profundidades de bits sean iguales. El codificador de vídeo puede aplicar además el proceso de transformación del espacio de color al codificar los datos de vídeo.

Esta divulgación describe técnicas que pueden mejorar el rendimiento de codificación de la transformación del espacio de color en bucle y pueden reducir la complejidad del decodificador. La FIG. 1 es un diagrama de bloques que ilustra un sistema 10 de codificación y decodificación de vídeo de ejemplo que puede utilizar técnicas para la codificación de contenido de pantalla, en el que se utiliza un formato de muestreo de croma alto. Como se muestra en la FIG. 1, el sistema 10 incluye un dispositivo 12 de origen que proporciona datos de vídeo codificados para ser decodificados más adelante por un dispositivo 14 de destino. En particular, el dispositivo 12 de origen proporciona los datos de vídeo al dispositivo 14 de destino a través de un medio 16 legible por ordenador. El dispositivo 12 de origen y el dispositivo 14 de destino pueden comprender cualquiera de una amplia gama de dispositivos, incluyendo ordenadores de escritorio, ordenadores portátiles (es decir, portátiles), tabletas, decodificadores, teléfonos tales como los llamados teléfonos "inteligentes", así llamados pads, televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivos de transmisión de vídeo o similares "inteligentes". En algunos casos, el dispositivo 12 de origen y el dispositivo 14 de destino pueden estar equipados para comunicación inalámbrica.

El dispositivo 14 de destino puede recibir los datos de vídeo codificados para ser decodificados a través de un medio 16 legible por ordenador. El medio 16 legible por ordenador puede comprender cualquier tipo de medio o dispositivo capaz de mover los datos de vídeo codificados desde el dispositivo 12 de origen al dispositivo 14 de destino. En un ejemplo, el medio 16 legible por ordenador puede comprender un medio de comunicación para permitir que el dispositivo 12 de origen transmita datos de vídeo codificados directamente al dispositivo 14 de destino en tiempo real. Los datos de vídeo codificados pueden modularse de acuerdo con un estándar de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitirse al dispositivo 14 de destino. El medio de comunicación puede comprender cualquier medio de comunicación inalámbrico o por cable, tal como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión físicas. El medio de comunicación puede formar parte de una red basada en paquetes, tal como una red de área local, una red de área amplia o una red global como Internet. El medio de comunicación puede incluir enrutadores, conmutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo 12 de origen al dispositivo 14 de destino.

En algunos ejemplos, los datos codificados pueden enviarse desde la interfaz 22 de salida a un dispositivo de almacenamiento. De manera similar, se puede acceder a los datos codificados desde el dispositivo de almacenamiento mediante una interfaz de entrada. El dispositivo de almacenamiento puede incluir cualquiera de una variedad de medios de almacenamiento de datos distribuidos o a los que se accede localmente, tales como un disco duro, discos Blu-ray, DVD, CD-ROM, memoria flash, memoria volátil o no volátil, o cualquier otro medio de almacenamiento digital adecuado para almacenar datos de vídeo codificados. En un ejemplo adicional, el dispositivo de almacenamiento puede corresponder a un servidor de archivos u otro dispositivo de almacenamiento intermedio que puede almacenar el vídeo codificado generado por el dispositivo 12 de origen. El dispositivo 14 de destino puede acceder a los datos de vídeo almacenados desde el dispositivo de almacenamiento mediante transmisión o descarga. El servidor de archivos puede ser cualquier tipo de servidor capaz de almacenar datos de vídeo codificados y transmitir esos datos de vídeo

codificados al dispositivo 14 de destino. Los servidores de archivos de ejemplo incluyen un servidor web (por ejemplo, para un sitio web), un servidor FTP, dispositivos de almacenamiento conectados a la red (NAS) o una unidad de disco local. El dispositivo 14 de destino puede acceder a los datos de vídeo codificados a través de cualquier conexión de datos estándar, incluyendo una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión Wi-Fi), una conexión por cable (por ejemplo, DSL, módem por cable, etc.) o una combinación de ambos que sea adecuada para acceder a datos de vídeo codificados almacenados en un servidor de archivos. La transmisión de datos de vídeo codificados desde el dispositivo de almacenamiento puede ser una transmisión de flujo continuo, una transmisión de descarga o una combinación de las mismas.

Las técnicas de esta divulgación no se limitan necesariamente a aplicaciones o configuraciones inalámbricas. Las técnicas se pueden aplicar a la codificación de vídeo como soporte de cualquiera de una variedad de aplicaciones multimedia, tales como transmisiones de televisión por aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de vídeo en transmisión por Internet, tales como transmisión adaptativa dinámica a través de HTTP (DASH), vídeo digital codificado en un medio de almacenamiento de datos, decodificación de vídeo digital almacenado en un medio de almacenamiento de datos u otras aplicaciones. En algunos ejemplos, el sistema 10 puede configurarse para admitir transmisión de vídeo unidireccional o bidireccional para admitir aplicaciones tales como transmisión de vídeo, reproducción de vídeo, transmisión de vídeo y/o videotelefonía.

En el ejemplo de la FIG. 1, el dispositivo 12 de origen incluye una fuente 18 de vídeo, un codificador 20 de vídeo y una interfaz 22 de salida. El dispositivo 14 de destino incluye la interfaz 28 de entrada, el decodificador 30 de vídeo y el dispositivo 32 de visualización. De acuerdo con esta divulgación, el codificador 20 de vídeo del dispositivo 12 de origen puede configurarse para aplicar las técnicas para codificar bloques de vídeo usando un proceso de conversión de espacio de color. En otros ejemplos, un dispositivo de origen y un dispositivo de destino pueden incluir otros componentes o disposiciones. Por ejemplo, el dispositivo 12 de origen puede recibir datos de vídeo desde una fuente 18 de vídeo externa, tal como una cámara externa. Asimismo, el dispositivo 14 de destino puede interactuar con un dispositivo de visualización externo, en lugar de incluir un dispositivo de visualización integrado.

El sistema 10 ilustrado de la FIG. 1 es simplemente un ejemplo. Las técnicas para codificar bloques de vídeo utilizando un proceso de conversión de espacio de color pueden realizarse mediante cualquier dispositivo de codificación y/o decodificación de vídeo digital. Aunque generalmente las técnicas de esta divulgación se realizan mediante un dispositivo de codificación de vídeo, las técnicas también se pueden realizar mediante un codificador/decodificador de vídeo, normalmente denominado "CODEC". Además, las técnicas de esta divulgación también pueden realizarse mediante un preprocesador de vídeo. El dispositivo 12 de origen y el dispositivo 14 de destino son simplemente ejemplos de dichos dispositivos de codificación en los que el dispositivo 12 de origen genera datos de vídeo codificados para su transmisión al dispositivo 14 de destino. En algunos ejemplos, los dispositivos 12, 14 pueden funcionar de una manera sustancialmente simétrica de modo que cada uno de los dispositivos 12, 14 incluya componentes de codificación y decodificación de vídeo. Por lo tanto, el sistema 10 puede soportar transmisión de vídeo unidireccional o bidireccional entre dispositivos 12, 14 de vídeo, por ejemplo, para transmisión de vídeo, reproducción de vídeo, transmisión de vídeo o videotelefonía.

La fuente 18 de vídeo del dispositivo 12 de origen puede incluir un dispositivo de captura de vídeo, tal como una cámara de vídeo, un archivo de vídeo que contiene vídeo capturado previamente y/o una interfaz de alimentación de vídeo para recibir vídeo de un proveedor de contenido de vídeo. Como alternativa adicional, la fuente 18 de vídeo puede generar datos basados en gráficos por ordenador como vídeo de origen, o una combinación de vídeo en directo, vídeo archivado y vídeo generado por ordenador. En algunos casos, si la fuente 18 de vídeo es una cámara de vídeo, el dispositivo 12 de origen y el dispositivo 14 de destino pueden formar los llamados teléfonos con cámara o teléfonos de vídeo. Sin embargo, como se mencionó anteriormente, las técnicas descritas en esta divulgación pueden ser aplicables a la codificación de vídeo en general y pueden aplicarse a aplicaciones inalámbricas y/o por cable. En cada caso, el vídeo capturado, precapturado o generado por ordenador puede codificarse mediante el codificador 20 de vídeo. La información de vídeo codificada puede entonces enviarse mediante la interfaz 22 de salida a un medio 16 legible por ordenador.

El medio 16 legible por ordenador puede incluir medios transitorios, tales como una transmisión inalámbrica o una transmisión de red por cable, o medios de almacenamiento (es decir, medios de almacenamiento no transitorios), tales como un disco duro, una unidad flash, un disco compacto, un disco de vídeo digital, disco Blu-ray u otro medio legible por ordenador. En algunos ejemplos, un servidor de red (no mostrado) puede recibir datos de vídeo codificados desde el dispositivo 12 de origen y proporcionar los datos de vídeo codificados al dispositivo 14 de destino, por ejemplo, mediante transmisión de red. De manera similar, un dispositivo informático de una instalación de producción de medios, tal como una instalación de estampado de discos, puede recibir datos de vídeo codificados desde el dispositivo 12 de origen y producir un disco que contenga los datos de vídeo codificados. Por lo tanto, se puede entender que el medio 16 legible por ordenador incluye uno o más medios legibles por ordenador de diversas formas, en diversos ejemplos.

La interfaz 28 de entrada del dispositivo 14 de destino recibe información del medio 16 legible por ordenador. La información del medio 16 legible por ordenador puede incluir información de sintaxis definida por el codificador 20 de vídeo, que también es utilizada por el decodificador 30 de vídeo, que incluye elementos de sintaxis que describen

características y/o procesamiento de bloques y otras unidades codificadas, por ejemplo, GOP. El dispositivo 32 de visualización muestra los datos de vídeo decodificados a un usuario, y puede comprender cualquiera de una variedad de dispositivos de visualización tales como un tubo de rayos catódicos (CRT), una pantalla de cristal líquido (LCD), una pantalla de plasma, un diodo orgánico emisor de luz (OLED) u otro tipo de dispositivo de visualización.

El codificador 20 de vídeo y el decodificador 30 de vídeo pueden funcionar de acuerdo con un estándar de codificación de vídeo, tal como el estándar de codificación de vídeo de alta eficiencia (HEVC) actualmente en desarrollo, y pueden ajustarse al modelo de prueba HEVC (HM). Alternativamente, el codificador 20 de vídeo y el decodificador 30 de vídeo pueden funcionar de acuerdo con otros estándares propietarios o industriales, tales como el estándar ITU-T H.264, denominado alternativamente MPEG-4, Parte 10, Codificación de Vídeo Avanzada (AVC), o extensiones de tales estándares. Sin embargo, las técnicas de esta divulgación no se limitan a ningún estándar de codificación particular. Otros ejemplos de estándares de codificación de vídeo incluyen MPEG-2 y ITU-T H.263. Aunque no se muestra en la FIG. 1, en algunos aspectos, el codificador 20 de vídeo y el decodificador 30 de vídeo pueden integrarse cada uno con un codificador y descodificador de audio, y pueden incluir unidades MUX-DEMUX apropiadas, u otro hardware y software, para manejar la codificación de audio y vídeo en un flujo de datos común o flujos de datos separados. Si corresponde, las unidades MUX-DEMUX pueden cumplir con el protocolo multiplexor ITU H.223 u otros protocolos como el protocolo de datagramas de usuario (UDP).

El estándar ITU-T H.264/MPEG-4 (AVC) fue formulado por el Grupo de Expertos en Codificación de Vídeo (VCEG) del ITU-T junto con el Grupo de Expertos en Imágenes en Movimiento (MPEG) de ISO/IEC como producto de una asociación colectiva conocida como el Equipo Conjunto de Vídeo (JVT). En algunos aspectos, las técnicas descritas en esta divulgación se pueden aplicar a dispositivos que generalmente cumplen con el estándar H.264. El estándar H.264 se describe en la Recomendación ITU-T H.264, Codificación de Vídeo Avanzada para servicios audiovisuales genéricos, del Grupo de Estudio del ITU-T, con fecha de marzo de 2005, a la que en el presente documento se puede hacer referencia como estándar H.264, o la especificación H.264, o el estándar o especificación H.264/AVC. El Equipo Conjunto de Vídeo (JVT) continúa trabajando en extensiones para H.264/MPEG-4 AVC.

El codificador 20 de vídeo y el decodificador 30 de vídeo pueden implementarse cada uno como cualquiera de una variedad de circuitos codificadores adecuados, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados de aplicación específica (ASIC), conjuntos de puertas programables en campo (FPGA), lógica discreta, software, hardware, firmware o cualquier combinación de los mismos. Cuando las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio adecuado, no transitorio, legible por ordenador y ejecutar las instrucciones en hardware usando uno o más procesadores para realizar las técnicas de esta divulgación. Cada uno de los codificadores 20 de vídeo y decodificadores 30 de vídeo puede incluirse en uno o más codificadores o decodificadores, cualquiera de los cuales puede integrarse como parte de un codificador/descodificador combinado (CODEC) en un dispositivo respectivo.

El JCT-VC está trabajando en el desarrollo del estándar HEVC. Los esfuerzos de estandarización de HEVC se basan en un modelo en evolución de codificador de vídeo denominado Modelo de Prueba HEVC (HM). El HM supone varias capacidades adicionales de los codificadores de vídeo en relación con los dispositivos existentes de acuerdo con, por ejemplo, ITU-T H.264/AVC. Por ejemplo, mientras que H.264 proporciona nueve modos de codificación predicción intra, el HM puede proporcionar hasta treinta y tres modos de codificación predicción intra.

En general, el modelo de trabajo del HM describe que una trama de vídeo o una imagen se puede dividir en una secuencia de bloques de árbol o unidades de codificación más grandes (LCU) que incluyen muestras de luma y croma. Los datos de sintaxis dentro de un flujo de bits pueden definir un tamaño para la LCU, que es la unidad de codificación más grande en términos de número de píxeles. Un segmento incluye una cantidad de bloques de árbol consecutivos en orden de codificación. Una trama de vídeo o una imagen se puede dividir en una o más porciones. Cada bloque de árbol se puede dividir en unidades de codificación (CU) de acuerdo con un árbol cuádruple. En general, una estructura de datos de cuatro árboles incluye un nodo por CU, con un nodo raíz correspondiente al bloque de árbol. Si una CU se divide en cuatro subCU, el nodo correspondiente a la CU incluye cuatro nodos hoja, cada uno de los cuales corresponde a una de las subCU.

Cada nodo de la estructura de datos de árbol cuádruple puede proporcionar datos de sintaxis para la CU correspondiente. Por ejemplo, un nodo en el árbol cuádruple puede incluir un indicador de división, que indica si la CU correspondiente al nodo está dividida en subCU. Los elementos de sintaxis para una CU pueden definirse de forma recursiva y pueden depender de si la CU está dividida en subCU. Si una CU no se divide más, se la denomina CU hoja. En esta divulgación, también se hará referencia a cuatro subCU de una CU de hoja como CU de hoja incluso si no hay una división explícita de la CU de hoja original. Por ejemplo, si una CU de tamaño 16x16 no se divide más, las cuatro subCU de 8x8 también se denominarán CU de hoja, aunque la CU de 16x16 nunca se dividió.

Una CU tiene un propósito similar al de un macrobloque del estándar H.264, excepto que una CU no tiene una distinción de tamaño. Por ejemplo, un bloque de árbol puede dividirse en cuatro nodos secundarios (también denominados subCU) y cada nodo secundario puede, a su vez, ser un nodo principal y dividirse en otros cuatro nodos secundarios. Un nodo hijo final no dividido, denominado nodo hoja del árbol cuádruple, comprende un nodo de codificación, también denominado CU hoja. Los datos de sintaxis asociados con un flujo de bits codificado pueden

definir un número máximo de veces que se puede dividir un bloque de árbol, denominado profundidad máxima de CU, y también pueden definir un tamaño mínimo de los nodos de codificación. En consecuencia, un flujo de bits también puede definir una unidad de codificación más pequeña (SCU). Esta divulgación utiliza el término "bloque" para referirse a cualquiera de una CU, PU o TU, en el contexto de HEVC, o estructuras de datos similares en el contexto de otros estándares (por ejemplo, macrobloques y subbloques de los mismos en H.264/AVC).

Una CU incluye un nodo de codificación y unidades de predicción (PU) y unidades de transformación (TU) asociadas con el nodo de codificación. Un tamaño de la CU corresponde a un tamaño del nodo de codificación y debe tener forma cuadrada. El tamaño de la CU puede variar desde 8x8 píxeles hasta el tamaño del bloque de árbol con un máximo de 64x64 píxeles o más. Cada CU puede contener una o más PU y una o más TU. Los datos de sintaxis asociados con una CU pueden describir, por ejemplo, la partición de la CU en una o más PU. Los modos de partición pueden diferir entre si la CU está codificada en modo directo o de salto, codificada en modo predicción intra o codificada en modo predicción inter. Las PU pueden dividirse para que no tengan forma cuadrada. Los datos de sintaxis asociados con una CU también pueden describir, por ejemplo, la partición de la CU en una o más TU de acuerdo con un árbol cuádruple. Una TU puede tener forma cuadrada o no cuadrada (por ejemplo, rectangular).

El estándar HEVC permite transformaciones de acuerdo con las TU, que pueden ser diferentes para diferentes CU. El tamaño de las TU normalmente se basa en el tamaño de las PU dentro de una CU determinada definida para una LCU particionada, aunque puede que este no sea siempre el caso. Las TU suelen ser del mismo tamaño o más pequeñas que las PU. En algunos ejemplos, las muestras residuales correspondientes a una CU se pueden subdividir en unidades más pequeñas utilizando una estructura de árbol cuádruple conocida como "árbol cuádruple residual" (RQT). Los nodos hoja del RQT pueden denominarse unidades de transformación (TU). Los valores de diferencia de píxeles asociados con las TU pueden transformarse para producir coeficientes de transformación, que pueden cuantificarse.

Una CU hoja puede incluir una o más unidades de predicción (PU). En general, una PU representa un área espacial correspondiente a toda o una parte de la CU correspondiente, y puede incluir datos para recuperar una muestra de referencia para la PU. Además, una PU incluye datos relacionados con la predicción. Por ejemplo, cuando la PU está codificada modo intra, los datos para la PU pueden incluirse en un árbol cuádruple residual (RQT), que puede incluir datos que describen un modo predicción intra para una TU correspondiente a la PU. Como otro ejemplo, cuando la PU está codificada entre modos, la PU puede incluir datos que definen uno o más vectores de movimiento para la PU. Los datos que definen el vector de movimiento para una PU pueden describir, por ejemplo, un componente horizontal del vector de movimiento, un componente vertical del vector de movimiento, una resolución para el vector de movimiento (por ejemplo, precisión de un cuarto de píxel o precisión de un octavo de píxel), una imagen de referencia a la que apunta el vector de movimiento, y/o una lista de imágenes de referencia (por ejemplo, Lista 0, Lista 1 o Lista C) para el vector de movimiento.

Una CU hoja que tiene una o más PU también puede incluir una o más unidades de transformación (TU). Las unidades de transformación se pueden especificar usando un RQT (también denominado estructura de árbol cuádruple TU), como se analizó anteriormente. Por ejemplo, un indicador de división puede indicar si una CU de hoja está dividida en cuatro unidades de transformación. Luego, cada unidad de transformación puede dividirse aún más en más subTU. Cuando una TU no se divide más, se la puede denominar TU hoja. Generalmente, para la codificación intra, todas las TU hoja que pertenecen a una CU hoja comparten el mismo modo de predicción intra. Es decir, generalmente se aplica el mismo modo de predicción intra para calcular los valores predichos para todas las TU de una CU hoja. Para la codificación intra, un codificador de vídeo puede calcular un valor residual para cada TU hoja usando el modo de predicción intra, como una diferencia entre la porción de la CU correspondiente a la TU y el bloque original. Una TU no está necesariamente limitada al tamaño de una PU. Por tanto, las TU pueden ser más grandes o más pequeñas que una PU. Para codificación intra, una PU puede ubicarse junto con una TU hoja correspondiente para la misma CU. En algunos ejemplos, el tamaño máximo de una hoja-TU puede corresponder al tamaño de la hoja-CU correspondiente.

Además, las TU de las CU hoja también pueden estar asociadas con respectivas estructuras de datos de árboles cuádruples, denominados árboles cuádruples residuales (RQT). Es decir, una CU hoja puede incluir un árbol cuádruple que indica cómo se divide la CU hoja en TU. El nodo raíz de un árbol cuádruple TU generalmente corresponde a una CU hoja, mientras que el nodo raíz de un árbol cuádruple CU generalmente corresponde a un bloque de árbol (o LCU). Las TU del RQT que no están divididas se denominan TU hoja. En general, esta divulgación utiliza los términos CU y TU para referirse a CU hoja y TU hoja, respectivamente, a menos que se indique lo contrario.

Una secuencia de vídeo normalmente incluye una serie de tramas de vídeo o imágenes. Un grupo de imágenes (GOP) generalmente comprende una serie de una o más imágenes de vídeo. Un GOP puede incluir datos de sintaxis en un encabezado del GOP, un encabezado de una o más imágenes, o en cualquier otro lugar, que describa una serie de imágenes incluidas en el GOP. Cada segmento de una imagen puede incluir datos de sintaxis de segmento que describen un modo de codificación para el segmento respectivo. El codificador 20 de vídeo normalmente funciona en bloques de vídeo dentro de porciones de vídeo individuales para codificar los datos de vídeo. Un bloque de vídeo puede corresponder a un nodo de codificación dentro de una CU. Los bloques de vídeo pueden tener tamaños fijos o variables, y pueden diferir en tamaño de acuerdo con un estándar de codificación específico.

Por ejemplo, el HM admite la predicción en varios tamaños de PU. Suponiendo que el tamaño de una CU particular es $2N \times 2N$, el HM admite la predicción intra en tamaños de PU de $2N \times 2N$ o $N \times N$, y la predicción inter en tamaños de PU simétricos de $2N \times 2N$, $2N \times N$, $N \times 2N$ o $N \times N$. El HM también admite particiones asimétricas para predicción inter en tamaños de PU de $2N \times nU$, $2N \times nD$, $nL \times 2N$ y $nR \times 2N$. En la partición asimétrica, una dirección de una CU no se divide, mientras que la otra dirección se divide en 25% y 75%. La porción de la CU correspondiente a la partición del 25% se indica con una "n" seguida de una indicación de "Arriba", "Abajo", "Izquierda" o "Derecha". Así, por ejemplo, " $2N \times nU$ " se refiere a una CU $2N \times 2N$ que está dividida horizontalmente con una PU $2N \times 0.5N$ en la parte superior y una PU $2N \times 1.5N$ en la parte inferior.

En esta divulgación, " $N \times N$ " y " N por N " se pueden usar indistintamente para referirse a las dimensiones de píxeles de un bloque de vídeo en términos de dimensiones verticales y horizontales, por ejemplo, 16×16 píxeles o 16 por 16 píxeles. En general, un bloque de 16×16 tendrá 16 píxeles en dirección vertical ($y = 16$) y 16 píxeles en dirección horizontal ($x = 16$). Del mismo modo, un bloque $N \times N$ generalmente tiene N píxeles en dirección vertical y N píxeles en dirección horizontal, donde N representa un valor entero no negativo. Los píxeles de un bloque se pueden organizar en filas y columnas. Además, los bloques no necesitan tener necesariamente el mismo número de píxeles en dirección horizontal que en dirección vertical. Por ejemplo, los bloques pueden comprender $N \times M$ píxeles, donde M no es necesariamente igual a N .

Después de la codificación predictiva intra o predictiva inter usando las PU de una CU, el codificador 20 de vídeo puede calcular datos residuales para las TU de la CU. Las PU pueden comprender datos de sintaxis que describen un método o modo de generar datos de píxeles predictivos en el dominio espacial (también denominado dominio de píxeles) y las TU pueden comprender coeficientes en el dominio de transformación después de la aplicación de una transformación, por ejemplo, un coseno discreto (DCT), una transformación entera, una transformación ondícula o una transformación conceptualmente similar a los datos de vídeo residuales. Los datos residuales pueden corresponder a diferencias de píxeles entre píxeles de la imagen no codificada y valores de predicción correspondientes a las PU. El codificador 20 de vídeo puede formar las TU que incluyen los datos residuales para la CU y luego transformar las TU para producir coeficientes de transformación para la CU.

Después de cualquier transformación para producir coeficientes de transformación, el codificador 20 de vídeo puede realizar la cuantificación de los coeficientes de transformación. La cuantificación generalmente se refiere a un proceso en el que los coeficientes de transformación se cuantifican para posiblemente reducir la cantidad de datos utilizados para representar los coeficientes, proporcionando una mayor compresión. El proceso de cuantificación puede reducir la profundidad de bits asociada con algunos o todos los coeficientes. Por ejemplo, un valor de n bits puede redondearse hacia abajo a un valor de m bits durante la cuantificación, donde n es mayor que m .

Después de la cuantificación, el codificador de vídeo puede explorar los coeficientes de transformación, produciendo un vector unidimensional a partir de la matriz bidimensional que incluye los coeficientes de transformación cuantificados. La exploración puede diseñarse para colocar coeficientes de mayor energía (y por lo tanto de menor frecuencia) en la parte frontal de la matriz y para colocar coeficientes de menor energía (y por lo tanto de mayor frecuencia) en la parte posterior de la matriz. En algunos ejemplos, el codificador 20 de vídeo puede utilizar un orden de exploración predefinido para explorar los coeficientes de transformación cuantificados para producir un vector serializado que puede codificarse por entropía. En otros ejemplos, el codificador 20 de vídeo puede realizar una exploración adaptativa. Después de explorar los coeficientes de transformación cuantificados para formar un vector unidimensional, el codificador 20 de vídeo puede codificar por entropía el vector unidimensional, por ejemplo, de acuerdo con codificación de longitud variable adaptativa al contexto (CAVLC), codificación aritmética binaria adaptativa al contexto (CABAC), codificación aritmética binaria adaptativa al contexto (SBAC) basada en sintaxis, codificación de entropía de partición de intervalo de probabilidad (PIPE) u otra metodología de codificación de entropía. El codificador 20 de vídeo también puede codificar por entropía elementos de sintaxis asociados con los datos de vídeo codificados para su uso por el decodificador 30 de vídeo al decodificar los datos de vídeo.

Para realizar CABAC, el codificador 20 de vídeo puede asignar un contexto dentro de un modelo de contexto a un símbolo que se va a transmitir. El contexto puede estar relacionado, por ejemplo, con si los valores vecinos del símbolo son distintos de cero o no. Para realizar CAVLC, el codificador 20 de vídeo puede seleccionar un código de longitud variable para que se transmita un símbolo. Las palabras de código en VLC pueden construirse de manera que códigos relativamente más cortos correspondan a símbolos más probables, mientras que códigos más largos correspondan a símbolos menos probables. De esta manera, el uso de VLC puede lograr un ahorro de bits respecto, por ejemplo, al uso de palabras de código de igual longitud para cada símbolo que se va a transmitir. La determinación de probabilidad puede basarse en un contexto asignado al símbolo.

En algunos ejemplos, un codificador de vídeo, tal como un codificador 20 de vídeo o un decodificador 30 de vídeo, puede realizar una o más técnicas de acuerdo con esta divulgación. Por ejemplo, un codificador de vídeo puede determinar una profundidad de bits de un componente luma de los datos de vídeo y una profundidad de bits de un componente croma de los datos de vídeo. En respuesta a que la profundidad de bits del componente luma sea diferente de la profundidad de bits del componente croma, el codificador de vídeo puede modificar una o ambas de la profundidad de bits del componente luma y la profundidad de bits del componente cromático de modo que las

profundidades de bits sean iguales. El codificador de vídeo puede aplicar además el proceso de transformación del espacio de color al codificar los datos de vídeo.

- 5 Un codificador de vídeo, tal como un codificador 2D de vídeo o un decodificador 3D de vídeo, puede realizar cualquiera de las técnicas descritas en relación con las FIGS. 1-8. Modificando las profundidades de bits de un componente luma y/o un componente croma, un decodificador de vídeo puede ser capaz de decodificar datos de vídeo de manera más eficiente cuando se utiliza la transformación del espacio de color, aumentando así la eficiencia de codificación general del sistema y reduciendo el consumo de energía.
- 10 Se puede utilizar una variedad de elementos de sintaxis en alineación con las técnicas de la presente divulgación. Estos elementos de sintaxis pueden incluir:

Tabla 1

seq_parameter_set_rbsp() {	Descriptor
sps_video_parameter_set_id	u(4)
sps_max_sub_layers_minus1	u(3)
sps_temporal_id_nesting_flag	u(1)
profile_tier_level(sps_max_sub_layers_minus1)	
...	
vui_parameters_present_flag	u(1)
if(vui_parameters_present_flag)	
vui_parameters()	
sps_extension_present_flag	u(1)
if(sps_extension_present_flag) {	
for(i = 0; i < 1; i++)	
sps_extension_flag[i]	u(1)
sps_extension_7bits	u(7)
if(sps_extension_flag[0]) {	
transform_skip_rotation_enabled_flag	u(1)
transform_skip_context_enabled_flag	u(1)
intra_block_copy_enabled_flag	u(1)
implicit_rdpcm_enabled_flag	u(1)
explicit_rdpcm_enabled_flag	u(1)
extended_precision_processing_flag	u(1)
intra_smoothing_disabled_flag	u(1)
high_precision_offsets_enabled_flag	u(1)
fast_rice_adaptation_enabled_flag	u(1)
cabac_bypass_alignment_enabled_flag	u(1)
color_transform_enabled_flag (new)	u(1)
}	
if(sps_extension_7bits)	
while(more_rbsp_data())	

(continuación)

sps_extension_data_flag	u(1)
}	
rbsp_trailing_bits()	
}	
coding_unit(x0, y0, log2CbSize) {	Descriptor
if(transquant_bypass_enabled_flag)	
cu_transquant_bypass_flag	ae(v)
if(slice_type != I)	
cu_skip_flag[x0][y0]	ae(v)
nCbs = (1 << log2CbSize)	
if(cu_skip_flag[x0][y0])	
prediction_unit(x0, y0, nCbs, nCbs)	
else {	
if(intra_block_copy_enabled_flag)	
intra_bc_flag[x0][y0]	ae(v)
if(slice_type != I && !intra_bc_flag[x0][y0])	
pred_mode_flag	ae(v)
if(CuPredMode[x0][y0] != MODE_INTRA intra_bc_flag[x0][y0] log2CbSize == MinCbLog2SizeY)	
part_mode	ae(v)
if(CuPredMode[x0][y0] == MODE_INTRA) {	
if(PartMode == PART_2Nx2N && pcm_enabled_flag && !intra_bc_flag[x0][y0] && log2CbSize >= Log2MinIpcmCbSizeY && log2CbSize <= Log2MaxIpcmCbSizeY)	
pcm_flag[x0][y0]	ae(v)
if(pcm_flag[x0][y0]) {	
while(!byte_aligned())	
pcm_alignment_zero_bit	f(1)
pcm_sample(x0, y0, log2CbSize)	
} else if(intra_bc_flag[x0][y0]) {	
mvd_coding(x0, y0, 2)	
if(PartMode == PART_2NxN)	
mvd_coding(x0, y0 + (nCbs / 2), 2)	
else if(PartMode == PART_Nx2N)	
mvd_coding(x0 + (nCbs / 2), y0, 2)	
else if(PartMode == PART_NxN) {	
mvd_coding(x0 + (nCbs / 2), y0, 2)	
mvd_coding(x0, y0 + (nCbs / 2), 2)	

(continuación)

mvd_coding(x0 + (nCbS / 2), y0 + (nCbS / 2), 2)	
}	
} else {	
pbOffset = (PartMode == PART_NxN) ? (nCbS / 2) : nCbS	
for(j = 0; j < nCbS; j = j + pbOffset)	
for(i = 0; i < nCbS; i = i + pbOffset)	
prev_intra_luma_pred_flag [x0 + i][y0 + j]	ae(v)
for(j = 0; j < nCbS; j = j + pbOffset)	
for(i = 0; i < nCbS; i = i + pbOffset)	
if(prev_intra_luma_pred_flag[x0 + i][y0 + j])	
mpm_idx [x0 + i][y0 + j]	ae(v)
else	
rem_intra_luma_pred_mode [x0 + i][y0 + j]	ae(v)
if(ChromaArrayType == 3)	
for(j = 0; j < nCbS; j = j + pbOffset)	
for(i = 0; i < nCbS; i = i + pbOffset)	
intra_chroma_pred_mode [x0 + i][y0 + j]	ae(v)
else if(ChromaArrayType != 0)	
intra_chroma_pred_mode [x0][y0]	ae(v)
}	
} else {	
...	
}	
if(!pcm_flag[x0][y0]) {	
if(CuPredMode[x0][y0] != MODE_INTRA && !(PartMode == PART_2Nx2N && merge_flag[x0][y0]) (CuPredMode[x0][y0] == MODE_INTRA && intra_beflag[x0][y0]))	
rqt_root_cbf	ae(v)
if(rqt_root_cbf) {	
if(color_transform_enabled_flag) (new)	
color_transform_flag [x0][y0] (new)	ae(v)
MaxTrafoDepth = (CuPredMode[x0][y0] == MODE_INTRA ? (max_transform_hierarchy_depth_intra + IntraSplitFlag) : max_transform_hierarchy_depth_inter)	
transform_tree(x0, y0, x0, y0, log2CbSize, 0, 0)	
}	
}	
}	
}	

Cuando el elemento de sintaxis `color_transform_enabled_flag` es igual a 1, se puede invocar el proceso de transformación del espacio de color en bucle en el proceso de decodificación. Cuando el elemento de sintaxis `color_transform_enabled_flag` es igual a 0, es posible que no se aplique la transformación del espacio de color en bucle. Cuando el elemento de sintaxis no está presente, se puede inferir que el valor del color del elemento de sintaxis `transform_enabled_flag` es igual a 0.

Cuando el elemento de sintaxis `color_transform_flag[x0][y0]` es igual a 1, la unidad de codificación actual puede codificarse con una transformación de espacio de color. Cuando el elemento de sintaxis `color_transform_flag[x0][y0]` es igual a 0, la unidad de codificación actual se codifica sin transformación de espacio de color. Cuando el elemento de sintaxis no está presente, se puede inferir que el valor del color del elemento de sintaxis `transform_flag` es igual a 0. Los índices de matriz `x0` e `y0` pueden especificar la ubicación (`x0`, `y0`) de una muestra de luma superior izquierda del bloque de codificación considerado en relación con la muestra de luma superior izquierda de la imagen.

En algunos ejemplos, el codificador de vídeo, tal como el codificador 20 de vídeo o el decodificador 30 de vídeo, puede determinar si se utiliza la conversión del espacio de color para una unidad de codificación. El codificador de vídeo puede establecer un valor de un elemento de sintaxis de la unidad de codificación para indicar el uso de la conversión de espacio de color. El codificador de vídeo puede aplicar una matriz de transformación de espacio de color a un bloque residual de la unidad de codificación. El codificador de vídeo puede decodificar el elemento de sintaxis de la unidad de codificación, en el que el elemento de sintaxis indica si la unidad de codificación se codificó usando conversión de espacio de color. El codificador de vídeo puede determinar si un valor del elemento de sintaxis indica que la unidad de codificación se codificó usando conversión de espacio de color. El codificador de vídeo puede aplicar una matriz de transformación inversa del espacio de color a un bloque residual de la unidad de codificación en respuesta a determinar que el elemento de sintaxis indica que la unidad de codificación se codificó usando conversión de espacio de color.

En un ejemplo, para CU codificadas intra, se puede señalar directamente una bandera. Alternativamente, para las CU codificadas intra, se puede señalar una bandera sólo cuando la CU no está en modo I-PCM. En otro ejemplo, para CU codificada inter y/o CU codificadas intra BC, se puede señalar un indicador solo cuando hay coeficientes distintos de cero en la CU actual, es decir, `rqt_root_cbf` es igual a 1. En otro ejemplo, la bandera no se señala cuando la CU está codificada en modo paleta. En todos los ejemplos anteriores, cuando el indicador de transformación de color decodificado es igual a 0, o el indicador no está presente en el flujo de bits para una CU, se puede omitir el proceso de transformación del espacio de color.

Además, se puede señalar un indicador en cualquier conjunto de parámetros de secuencia (SPS)/conjunto de parámetros de imagen (PPS)/encabezado de segmento para controlar el uso de la transformación del espacio de color. Cuando el indicador en el encabezado SPS/PPS/segmento es igual a 0, la señalización del indicador de nivel de CU puede omitirse en todas las CU en la secuencia/imagen/segmento correspondiente, respectivamente. Alternativamente, la transformación del espacio de color se puede realizar en el nivel de PU o en el nivel de TU, en donde cada PU/TU tiene un indicador para indicar el uso de la transformación del espacio de color.

En un ejemplo, se puede aplicar una transformación YCoCg modificada en el proceso de codificación y decodificación. La transformación YCoCg modificada puede definirse de la siguiente manera:

Adelante:

$$\begin{bmatrix} Y \\ Co \\ Cg \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 0 & -2 \\ -1 & 2 & -1 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix},$$

Inversa:

$$G = (Y - Co + Cg + \text{offset}) \gg 2;$$

$$B = (\text{temp} - Co + \text{offset}) \gg 2;$$

$$\text{temp} = Y - Cg;$$

$R = (\text{compensación de } Co \text{ temporal}) \gg 2$; donde la compensación es igual a 2.

Alternativamente, la transformación YCoCg modificada se puede definir de la siguiente manera:

Adelante:

$$\begin{bmatrix} Y \\ Co \\ Cg \end{bmatrix} = \begin{bmatrix} 1/2 & 1 & 1/2 \\ 1 & 0 & -1 \\ -1/2 & 1 & -1/2 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Inversa:

$$G = (Y - Co + Cg + \text{offset}) \gg 1;$$

$$B = (\text{temp} - Co + \text{offset}) \gg 1;$$

$$R = (\text{temp} + Co + \text{offset}) \gg 1;$$

en donde el desplazamiento es igual a 1 o 0.

$$\text{temp} = Y - Cg;$$

Se puede inferir que el parámetro de cuantificación correspondiente utilizado por las CU o bloques que están codificados con transformación de color es igual a (dQP+2), mientras que se puede inferir que el utilizado por las CU o bloques sin transformación de color es igual a dQP. Al mismo tiempo, la profundidad de bits se puede aumentar aún más en 1 tanto en el proceso de cuantificación como en el de transformación.

Alternativamente, se aplican dos pasos de transformación hacia adelante/inversa con la transformación YCoCg original sin cambios más un proceso de normalización. Para la transformación directa, primero se aplica la transformación directa YCoCg original. Luego, para cada componente i, es decir, Y, Co y Cg, el componente se restablece a (i * forward_factor + offset) >> BIT_PRECISION donde BIT_PRECISION es un entero sin signo y forward_factor depende de BIT_PRECISION. En un ejemplo, forward_factor es igual a $1/\sqrt{6} * (1 \ll \text{BITS_TRANS}) + 0.5$, BIT_PRECISION es igual a 15, el desplazamiento es igual a $(1 \ll (\text{BIT_PRECISION} - 1))$.

Para la transformación inversa, primero se aplica la transformación inversa YCoCg original. Luego, para cada componente i, es decir, Y, Co y Cg, el componente se restablece a (i * backward_factor + offset) >> BIT_PRECISION en donde BIT_PRECISION es el mismo que se usa en la transformación directa y el backward_factor depende de BIT_PRECISION. En un ejemplo, el backward_factor es igual a $\sqrt{6}/4 * (1 \ll \text{BITS_TRANS}) + 0.5$, el desplazamiento es igual a $(1 \ll (\text{BIT_PRECISION} - 1))$.

Alternativamente/además, el indicador de nivel CU puede señalarse sólo cuando al menos uno de los indicadores de bloque codificados en tres componentes de color (es decir, cbf_luma, cbf_cb y cbf_cr) es igual a 1. Alternativamente, el residual modificado después de la transformación del espacio de color se puede modificar aún más para garantizar que el intervalo del residual en CU o bloques con transformación del espacio de color sea el mismo que el del residual en CU/bloques sin transformación del espacio de color. En un ejemplo, se aplica una operación de recorte.

La matriz de transformación del espacio de color puede ser independiente de los píxeles reconstruidos. En cambio, puede depender del modo de codificación con o sin pérdidas. En un ejemplo, cuando la CU se codifica con modo con pérdida, se puede aplicar YCoCg mientras que se aplica YCoCg-R cuando la CU se codifica con modo sin pérdidas. Además, cuando se aplica YCoCg-R, la profundidad de broca de los componentes Co y Cg puede aumentar en 1. En otro ejemplo, la matriz de transformación puede depender de los modos intra, inter/intra BC. En este caso, se puede especificar una matriz predefinida para cada modo tanto en el codificador como en el decodificador y se puede usar la misma matriz tanto en el codificador como en el decodificador de acuerdo con el modo de codificación cuando el indicador de nivel CU es 1. Además, el indicador de nivel CU puede señalarse sólo cuando al menos uno de los indicadores de bloque codificados en tres componentes de color (es decir, cbf_luma, cbf_cb y cbf_cr) es igual a 1.

Las técnicas de la presente divulgación también pueden incluir un método de señalización y derivación de matriz de transformación, tal como a nivel de trama. En un ejemplo, para cada trama, se puede derivar una matriz de transformación basada en las estadísticas de los componentes de color de la imagen original. La matriz de transformación puede codificarse en un flujo de bits y transmitirse directamente al decodificador. Alternativamente, se puede aplicar la predicción de matriz de transformación entre tramas codificadas consecutivamente. Por ejemplo, se puede transmitir un indicador para indicar si la matriz es la misma que la utilizada en la trama anterior. Alternativamente, se podría derivar una matriz de transformación basada en un conjunto de tramas originales y señalizarla en el flujo de bits.

En otro ejemplo, se deriva y señala un conjunto de matrices de transformación en la sintaxis de alto nivel, tal como PPS. En un ejemplo, la imagen se clasifica en múltiples regiones de acuerdo con las características y, para cada región, se deriva una matriz de transformación. Las matrices derivadas se pueden codificar y señalizar en la sintaxis de alto nivel. El índice de la matriz de transformación seleccionada se indica además en el nivel CU/TU/PU. Alternativamente, el índice de la matriz de transformación seleccionada se señala en el encabezado del segmento o para cada región del mosaico. Alternativamente, la matriz de transformación se puede derivar en el encabezado del segmento o para cada región del mosaico.

Quando la profundidad de bits de los componentes luma y croma es diferente, la profundidad de bits de todos los componentes de color se puede modificar primero para que sea la misma y luego se puede aplicar la transformación de color. Alternativamente, el tren de bits puede ajustarse a una restricción de que, cuando la profundidad de bits de los componentes luma y croma sean diferentes en el tren de bits codificado, la transformación de color deberá inhabilitarse (es decir, `color_transform_enabled_flag` igual a 0).

Se pueden aplicar diferentes QP a CU/bloques que están codificados con o sin transformación de color, y/o la profundidad de bits utilizada en el proceso de cuantificación y transformación se puede modificar debido a dicha transformación de color no normalizada. En un ejemplo, cuando se utiliza la transformación YCoCg modificada anteriormente, los siguientes pasos (es decir, proceso de escalado, transformación y construcción de matriz antes del proceso de filtro de desbloqueo) se modifican aún más con base en `color_transform_flag`.

En un ejemplo, cuando se usa la transformación YCoCg modificada, como se describió anteriormente, el proceso de derivación del parámetro de cuantificación correspondiente y el proceso de transformación inversa se modifican en consecuencia. Las entradas a este proceso pueden incluir una ubicación de luma (`xCb`, `yCb`) que especifica la muestra superior izquierda del bloque de codificación de luma actual en relación con la muestra de luma superior izquierda de la imagen actual. En este proceso, se pueden derivar la variable `QpY`, el parámetro de cuantificación de luma `QpY` y los parámetros de cuantificación de croma `QpCb` y `QpCr`. La ubicación de luma (`xQg`, `yQg`) puede especificar la muestra de luma superior izquierda del grupo de cuantificación actual en relación con la muestra de luma superior izquierda de la imagen actual. Las posiciones horizontal y vertical `xQg` e `yQg` se pueden establecer iguales a $xCb - (xCb \& ((1 < \text{Log2MinCuQpDeltaSize}) - 1))$ e $yCb - (yCb \& ((1 < \text{Log2MinCuQpDeltaSize}) - 1))$, respectivamente. El tamaño de luma de un grupo de cuantificación, `Log2MinCuQpDeltaSize`, puede determinar el tamaño de luma del área más pequeña dentro de un bloque de árbol de codificación que comparte el mismo `qPY_PRED`.

La variable `QpY` se puede derivar de la siguiente manera:

$$QpY = ((qPY_PRED + CuQpDeltaVal + 52 + 2 * QpBdOffsetY) \% (52 + QpBdOffsetY)) - QpBdOffsetY$$

El parámetro de cuantificación de luma `QpY` puede derivarse de la siguiente manera:

$$Qp'Y = QpY + QpBdOffsetY$$

Quando el elemento de sintaxis `ChromaArrayType` no es igual a 0, se puede aplicar lo siguiente:
Las variables `qPiCb` y `qPiCr` se pueden derivar de la siguiente manera:

$$qPiCb = \text{Clip3}(-QpBdOffsetC, 57, QpY + pps_cb_qp_offset + slice_cb_qp_offset + CuQpAdjValCb)$$

$$qPiCr = \text{Clip3}(-QpBdOffsetC, 57, QpY + pps_cr_qp_offset + slice_cr_qp_offset + CuQpAdjValCr)$$

Si el elemento de sintaxis `ChromaArrayType` es igual a 1, las variables `qPCb` y `qPCr` se pueden establecer iguales al valor de `QpC` como se especifica en la siguiente tabla con base en el índice `qPi` igual a `qPiCb` y `qPiCr`, respectivamente. De lo contrario, las variables `qPCb` y `qPCr` se establecen en $\text{Min}(qPi, 51)$, con base en el índice `qPi` es igual a `qPiCb` y `qPiCr`, respectivamente.

Tabla 2

qPi	< 30	30	31	32	33	34	35	36	37	38	39	40	41	42	43	> 43
Qp _c	= qPi	29	30	31	32	33	33	34	34	35	35	36	36	37	37	= qPi - 6

Los parámetros de cuantificación de croma para los componentes Cb y Cr, `QpCb` y `QpCr`, pueden derivarse de la siguiente manera:

$$Qp'Cb = qPCb + QpBdOffsetC$$

$$Qp'Cr = qPCr + QpBdOffsetC$$

Cuando `color_transform_flag` es igual a 1, se puede aplicar lo siguiente:

$$Qp'Y = (Qp'Y + 8)$$

5 *When ChromaArrayType $\neq 0$, $Qp'Cb = (Qp'cb + 8)$ and $Qp'Cr = (Qp'cr + 8)$*

$$BitDepthY = (BitDepthY + 2)$$

$$BitDepthC = (BitDepthC + 2)$$

10 Las entradas a este proceso pueden incluir una ubicación de luma (`xTbY`, `yTbY`) que especifica la muestra superior izquierda del bloque de transformación de luma actual en relación con la muestra de luma superior izquierda de la imagen actual, una variable `nTbS` que especifica el tamaño del bloque de transformación actual, una variable `cldx` que especifica el componente de color del bloque actual y una variable `qP` que especifica el parámetro de cuantificación. El resultado de este proceso puede incluir la matriz `d` (`nTbS`)x(`nTbS`) de coeficientes de transformación escalados con elementos `d[x][y]`.

Las variables `log2TransformRange`, `bdShift`, `coeffMin` y `coeffMax` se pueden derivar de la siguiente manera:

When `color_transform_flag = 1`, the following applies:

$$CoeffCTMinY = -(1 << (extended_precision_processing_flag ? \\ Max(15, BitDepthY+6):15))$$

$$CoeffCTMinC = -(1 << (extended_precision_processing_flag ? \\ Max(15, BitDepthC+6):15))$$

$$CoeffCTMaxY = (1 << (extended_precision_processing_flag ? \\ Max(15, BitDepthY+6):15)) - 1$$

$$CoeffCTMaxC = (1 << (extended_precision_processing_flag ? \\ Max(15, BitDepthC+6):15)) - 1$$

If $cIdx = 0$,

$\log2TransformRange = extended_precision_processing_flag ?$

$Max(15, BitDepthY + 6) : 15$

$bdShift = BitDepthY + \log2(nTbS) + 10 - \log2TransformRange$

$coeffMin = (color_transform_flag ? CoeffCTMinY : CoeffMinY)$

$coeffMax = (color_transform_flag ? CoeffCTMaxY : CoeffMaxY)$

Otherwise,

$\log2TransformRange = extended_precision_processing_flag ?$

$Max(15, BitDepthC + 6) : 15$

$bdShift = BitDepthC + \log2(nTbS) + 10 - \log2TransformRange$

$coeffMin = (color_transform_flag ? CoeffCTMinC : CoeffMinC)$

$coeffMax = (color_transform_flag ? CoeffCTMaxC : CoeffMaxC)$

La lista $LevelScale[]$ se puede especificar como $LevelScale[k] = \{ 40, 45, 51, 57, 64, 72 \}$ con $k = 0..5$.

- 5 Para derivar los coeficientes de transformación escalada $d[x][y]$ con $x = 0..nTbS - 1$, $y = 0..nTbS - 1$, se puede aplicar lo siguiente:

El factor de escala $m[x][y]$ se puede derivar de la siguiente manera:

Si una o más de las siguientes condiciones son verdaderas, $m[x][y]$ se establece en 16:

- 10 $scaling_list_enabled_flag = 0$.
 $transform_skip_flag[xTbY][yTbY] = 1$ y $nTbS > 4$.
 De lo contrario se aplica lo siguiente:

$$m[x][y] = ScalingFactor[sizeId][matrixId][x][y] \quad (8-283)$$

- 15 Cuando se especifica el elemento de sintaxis $sizeId$ para el tamaño de la matriz de cuantificación igual a $(nTbS) \times (nTbS)$ y se especifica $MatrixId$ en la Tabla 7-4 para $sizeId$, $CuPredMode[xTbY][yTbY]$ y $cIdx$, respectivamente, entonces el coeficiente de transformación escalada $d[x][y]$ se puede derivar de la siguiente manera:

$$d[x][y] = Clip3(coeffMin, coeffMax, ((TransCoeffLevel[xTbY][yTbY][cIdx]$$

- 20 $[x][y] * m[x][y] * levelScale[qP \% 6] << (qP / 6)) + (1 << (bdShift - 1))) >> bdShift)$
 Cuando $color_transform_flag$ es igual a 1, se puede aplicar lo siguiente:

$$BitDepthY = BitDepthY - 2$$

$$25 \quad BitDepthC = BitDepthC - 2$$

- En un ejemplo, cuando se aplica la transformación del espacio de color a los modos intra, el residuo se deriva en el dominio del espacio de color convertido. Es decir, antes de invocar el proceso de predicción muestra intra, los píxeles vecinos de la CU actual pueden convertirse primero a otro espacio de color subóptimo con una transformación YCoCg o YCoCg-R directa modificada. Los píxeles vecinos modificados pueden usarse entonces para derivar el predictor de la CU actual. El residual (es decir, el error de predicción) se deriva de la CU actual y de los píxeles vecinos de la CU actual en el dominio subóptimo. El residual se codifica de la misma manera que el flujo de codificación convencional, como la predicción residual de componentes inter, la transformación, la cuantificación y la codificación de entropía. Después de la codificación de entropía, los coeficientes de transformación se pueden modificar aún más con la

transformación inversa YCoCg o YCoCg-R. Los coeficientes de transformación modificados pueden luego agregarse al predictor para derivar los píxeles reconstruidos de la CU actual. Después de invocar el proceso de reconstrucción, se puede aplicar la transformación inversa YCoCg o YCoCg-R a los píxeles vecinos modificados. Por lo tanto, en el lado del decodificador, antes de invocar el proceso de reconstrucción, se puede aplicar un proceso adicional en el que se aplica la transformación inversa del espacio de color a los coeficientes de transformación derivados.

Cuando se aplica la transformación del espacio de color a los modos intra, en lugar de generar el residuo restando el predictor del bloque actual en el dominio de color convertido, el residuo se puede generar usando los píxeles antes de la transformación del espacio de color, seguido de la transformación del espacio de color. En este caso, los modos intra luma y croma se pueden configurar en el mismo modo. Alternativamente, se puede omitir la señalización del modo croma. Alternativamente, el indicador de transformación de color puede señalarse sólo cuando los modos intra luma y croma son los mismos.

Como se describió anteriormente, en algunos ejemplos, los métodos de codificación sin pérdidas y con pérdidas pueden compartir la misma matriz, por ejemplo, YCoCg-R. Adicional o alternativamente, cuando la transformación YCoCg-R se aplica a la codificación con pérdida, la profundidad de bits de los componentes cromáticos de los bloques con transformación de color se puede incrementar en 1 en comparación con los bloques sin transformación de color. El componente QP luma de los bloques con transformación de color puede ser igual al de los bloques sin transformación de color menos 4. En un ejemplo, los dos componentes QP croma de bloques con transformación de color pueden ser iguales al componente QP croma/luma de bloques sin transformación de color más 2. En un ejemplo, los componentes QP croma Cg de bloques con transformación de color pueden ser iguales al componente QP de luma/croma de bloques sin transformación de color más 2. Los componentes QP croma Co de bloques con transformación de color pueden ser iguales al componente QP de luma/croma de bloques sin transformación de color más 3.

En tales ejemplos, cuando se utiliza YCoCg-R tanto para el modo con pérdida como para el modo sin pérdida, si el bloque actual está codificado con pérdida, la profundidad de bits de luma y croma puede ser la misma para aquellos bloques codificados sin habilitar la transformación de color adaptativa. En este caso, cuando se aplica codificación con pérdida, después de la transformación directa de YCoCg-R, los componentes Co y Cg se pueden escalar mediante un desplazamiento a la derecha de N bits para reducir el aumento de profundidad de bits debido a la transformación directa de YCoCg-R y hacer la precisión de profundidad de bits es la misma que la precisión de profundidad de bits del componente Y. Además, antes de realizar la transformación YCoCg-R inversa, los componentes Co y Cg pueden modificarse con un desplazamiento a la izquierda de N bits. En un ejemplo, N es igual a 1.

En un ejemplo, se puede suponer que la profundidad de bits de entrada de los componentes luma y croma es la misma. Para el modo de codificación con pérdidas, se pueden aplicar operaciones de desplazamiento a los dos componentes croma (es decir, Cg, Co) después de la transformación directa y antes de la transformación inversa. En un ejemplo, el siguiente proceso se aplica en orden:

1. El YCoCg-R directo no ha cambiado:

$$Co = R - B$$

$$t = B + \lfloor Co/2 \rfloor$$

$$Cg = G - t$$

$$Y = t + \lfloor Cg/2 \rfloor$$

2. Si el bloque actual está codificado en modo con pérdida, se puede aplicar lo siguiente:

$$C_g = C_g \gg 1;$$

$$C_o = C_o \gg 1;$$

5 En otro ejemplo, se puede considerar una compensación en el proceso de desplazamiento a la derecha. Por ejemplo, las ecuaciones resaltadas anteriormente podrían reemplazarse por:

$$C_g = (C_g + 1) \gg 1;$$

$$C_o = (C_o + 1) \gg 1;$$

Antes de invocar la transformación YCoCg-R inversa, puede aplicarse lo siguiente:
Si el bloque actual está codificado en modo con pérdida, se puede aplicar lo siguiente:

$$C_g = C_g \ll 1;$$

$$C_o = C_o \ll 1;$$

20 Y la YCoCg-R inversa puede permanecer sin cambios:

$$t = Y - [C_g/2]$$

$$G = C_g + t$$

$$B = t - [C_o/2]$$

$$R = B + C_o$$

30 En tales ejemplos, se puede invocar un proceso de modificación residual para bloques de transformación que utilizan una transformación de color adaptativa cuando ChromaArrayType es igual a 3. Las entradas a este proceso pueden incluir una variable blkSize que especifica el tamaño del bloque, una matriz (blkSize)x(blkSize) de muestras residuales de luma rY con elementos rY[x][y], una matriz (blkSize)x(blkSize) de muestras residuales de croma rCb con elementos rCb[x][y], y una matriz (blkSize)x(blkSize) de muestras residuales de croma rCr con elementos rCr[x][y]. Las salidas de este proceso pueden incluir una matriz (blkSize)x(blkSize) modificada rY de muestras residuales de luma, una matriz (blkSize)x(blkSize) modificada rCb de muestras residuales de croma y una matriz (blkSize)x(blkSize) modificada rCr de muestras de croma residual.

Las matrices (blkSize)x(blkSize) de muestras residuales rY, rCb y rCr con x = 0..blkSize - 1, y = 0..blkSize - 1 se puede modificar de la siguiente manera:

40 If cu_transquant_bypass_flag = 0, rCb[x][y] = rCb[x][y] << 1 and rCr[x][y] = rCr[x][y] << 1

$$tmp = r_Y[x][y] - (r_{Cb}[x][y] \gg 1)$$

$$r_Y[x][y] = tmp + r_{Cb}[x][y]$$

$$r_{Cb}[x][y] = tmp - (r_{Cr}[x][y] \gg 1)$$

$$r_{Cr}[x][y] = r_{Cb}[x][y] + r_{Cr}[x][y]$$

Alternativamente, 'si `cu_transquant_bypass_flag` es igual a 0, $r_{Cb}[x][y] = r_{Cb}[x][y] \ll 1$ and $r_{Cr}[x][y] = r_{Cr}[x][y] \ll 1$ ' podría reemplazarse por ' $r_{Cb}[x][y] = r_{Cb}[x][y] \ll (1 - cu_transquant_bypass_flag)$ $r_{Cr}[x][y] = r_{Cr}[x][y] \ll (1 - cu_transquant_bypass_flag)$ '.

5 En otro ejemplo de un método de codificación con pérdida y un método de codificación sin pérdida que comparten una matriz de transformación de color, la profundidad de bits de entrada de los componentes luma y croma puede ser diferente. En tal ejemplo, cuando las profundidades de bits de entrada de los componentes luma y croma son diferentes y se usa YCoCg-R para modos de codificación con y sin pérdidas, antes de la transformación inversa, al menos uno de los valores del componente Y, Co, Cg puede desplazarse primero N bits para hacer que los tres componentes
10 tengan la misma precisión de profundidad de bits. Además, se puede desplazar un bit más a los dos componentes croma (es decir, Co y Cg). Al menos uno de los valores de predicción de los tres componentes puede modificarse para que tenga la misma precisión de profundidad de bits antes de invocar la transformación YCoCg-R directa. Y después de la transformación directa, los dos componentes croma se pueden desplazar a la derecha 1 bit. Para los modos de codificación sin pérdidas, es posible que no se consideren las diferentes profundidades de bits y, por lo tanto, la transformación adaptativa del espacio de color puede estar deshabilitada. En un ejemplo, el flujo de bits puede incluir una restricción de que, cuando la profundidad de bits de los componentes luma y croma sean diferentes en el flujo de bits codificado, la transformación de color deberá desactivarse (es decir, `color_transform_enabled_flag` igual a 0).

En este ejemplo, se puede invocar un proceso de modificación residual para bloques de transformación que utilizan una transformación de color adaptativa cuando `ChromaArrayType` es igual a 3. Las entradas a este proceso pueden incluir una variable `blkSize` que especifica el tamaño del bloque, una matriz (`blkSize`)x(`blkSize`) de muestras residuales de luma r_Y con elementos $r_Y[x][y]$, una matriz (`blkSize`)x(`blkSize`) de muestras residuales de croma r_{Cb} con elementos $r_{Cb}[x][y]$, y una matriz (`blkSize`)x(`blkSize`) de muestras residuales de croma r_{Cr} con elementos $r_{Cr}[x][y]$. Las salidas de este proceso pueden incluir una matriz (`blkSize`)x(`blkSize`) modificada r_Y de muestras residuales de luma, una matriz (`blkSize`)x(`blkSize`) modificada r_{Cb} de muestras residuales de croma y una matriz (`blkSize`)x(`blkSize`) modificada r_{Cr} de muestras residuales de croma.

Las variables `deltaBDY` y `deltaBDC` puede derivarse de la siguiente manera:

30
$$\text{BitDepthMax} = \text{Max}(\text{BitDepthY}, \text{BitDepthC})$$

$$\text{deltaBDY} = \text{cu_transquant_bypass_flag} ? 0 : \text{BitDepthMax} - \text{BitDepthY}$$

$$\text{deltaBDC} = \text{cu_transquant_bypass_flag} ? 0 : \text{BitDepthMax} - \text{BitDepthC}$$

35
$$\text{OY} = \text{cu_transquant_bypass_flag} || (\text{BitDepthMax} == \text{BitDepthY}) ? 0 : 1 \ll (\text{deltaBDY} - 1)$$

$$\text{OC} = \text{cu_transquant_bypass_flag} || (\text{BitDepthMax} == \text{BitDepthC}) ? 0 : 1 \ll (\text{deltaBDC} - 1)$$

40 Las matrices (`blkSize`)x(`blkSize`) de muestras residuales r_Y , r_{Cb} y r_{Cr} con $x = 0..blkSize - 1$, $y = 0..blkSize - 1$ se puede modificar de la siguiente manera:
Las muestras residuales $r_Y[x][y]$, $r_{Cb}[x][y]$ y $r_{Cr}[x][y]$ podrán modificarse de la siguiente manera:

$$\begin{aligned} r_Y[x][y] &= r_Y[x][y] \ll \text{deltaBDY} \\ r_{Cb}[x][y] &= r_{Cb}[x][y] \ll (\text{deltaBDC} + 1 - \text{cu_transquant_bypass_flag}) \\ r_{Cr}[x][y] &= r_{Cr}[x][y] \ll (\text{deltaBDC} + 1 - \text{cu_transquant_bypass_flag}) \\ \text{tmp} &= r_Y[x][y] - (r_{Cb}[x][y] \gg 1) \\ r_Y[x][y] &= \text{tmp} + r_{Cb}[x][y] \\ r_{Cb}[x][y] &= \text{tmp} - (r_{Cr}[x][y] \gg 1) \\ r_{Cr}[x][y] &= r_{Cb}[x][y] + r_{Cr}[x][y] \\ r_Y[x][y] &= (r_Y[x][y] + \text{OY}) \gg \text{deltaBDY} \end{aligned}$$

$$rCb[x][y] = (rCb[x][y] + Oc) >> \text{deltaBDC}$$

$$rCr[x][y] = (rCr[x][y] + Oc) >> \text{deltaBDC}$$

5 En los ejemplos anteriores, la función Max se puede utilizar para derivar el valor mayor entre dos variables. Alternativamente, al menos uno de los desplazamientos de redondeo O_Y y O_C puede establecerse igual a 0.

10 Alternativamente, incluso para codificación sin pérdidas, cuando la profundidad de bits de los componentes luma y croma son diferentes, los componentes pueden desplazarse a la misma precisión de profundidad de bits, si es necesario. En el lado del codificador, después de la transformación directa, es posible que no haya necesidad de volver a la precisión de profundidad de bits original. En otras palabras, el residuo codificado de los tres componentes puede tener la misma profundidad de bits para garantizar que se trata de una codificación sin pérdidas. A diferencia de la codificación con pérdidas, las entradas no se pueden modificar antes de invocar la transformación YCoCg-R hacia atrás para la codificación sin pérdidas. Es posible que aún sea necesario realizar el desplazamiento hacia la derecha para garantizar que las salidas tengan la misma profundidad de bits que las entradas originales.

20 En este ejemplo, se puede invocar un proceso de modificación residual para bloques de transformación que utilizan una transformación de color adaptativa cuando ChromaArrayType es igual a 3. Las entradas a este proceso pueden incluir una variable blkSize que especifica el tamaño del bloque, una matriz (blkSize)x(blkSize) de muestras residuales de luma rY con elementos rY[x][y], una matriz (blkSize)x(blkSize) de muestras residuales de croma rCb con elementos rCb[x][y], y una matriz (blkSize)x(blkSize) de muestras residuales de croma rCr con elementos rCr[x][y]. Las salidas de este proceso pueden incluir una matriz (blkSize)x(blkSize) modificada rY de muestras residuales de luma, una matriz (blkSize)x(blkSize) modificada rCb de muestras residuales de croma y una matriz (blkSize)x(blkSize) modificada rCr de muestras residuales de croma.

25 Las variables deltaBDY y deltaBDC puede derivarse de la siguiente manera:

$$\text{BitDepthMax} = \text{Max}(\text{BitDepthY}, \text{BitDepthC})$$

$$30 \quad \text{deltaBDY} = \text{BitDepthMax} - \text{BitDepthY}$$

$$\text{deltaBDC} = \text{BitDepthMax} - \text{BitDepthC}$$

$$35 \quad OY = (\text{BitDepthMax} == \text{BitDepthY}) ? 0 : 1 << (\text{deltaBDY} - 1)$$

$$OC = (\text{BitDepthMax} == \text{BitDepthC}) ? 0 : 1 << (\text{deltaBDC} - 1)$$

Las matrices (blkSize)x(blkSize) de muestras residuales rY, rCb y rCr con $x = 0..blkSize - 1$, $y = 0..blkSize - 1$ se puede modificar de la siguiente manera:

40 Cuando el indicador de derivación cu_transquant es igual a 0, las muestras residuales rY[x][y], rCb[x][y] y rCr[x][y] pueden modificarse de la siguiente manera:

$$rY[x][y] = rY[x][y] << \text{deltaBDY}$$

$$rCb[x][y] = rCb[x][y] << (\text{deltaBDC} + 1 - \text{cu_transquant_bypass_flag})$$

$$rCr[x][y] = rCr[x][y] << (\text{deltaBDC} + 1 - \text{cu_transquant_bypass_flag})$$

$$\text{tmp} = rY[x][y] - (rCb[x][y] >> 1)$$

$$rY[x][y] = \text{tmp} + rCb[x][y]$$

$$rCb[x][y] = \text{tmp} - (rCr[x][y] >> 1)$$

$$rCr[x][y] = rCb[x][y] + rCr[x][y]$$

$$rY[x][y] = (rY[x][y] + OY) >> \text{deltaBDY}$$

$$rCb[x][y] = (rCb[x][y] + O_c) \gg \text{deltaBDC}$$

$$rCr[x][y] = (rCr[x][y] + O_c) \gg \text{deltaBDC}$$

5 Alternativamente, uno de O_Y y O_C puede ser igual a 0.

El codificador 20 de vídeo puede enviar además datos de sintaxis, tales como datos de sintaxis basados en bloques, datos de sintaxis basados en tramas y datos de sintaxis basados en GOP, al decodificador 30 de vídeo, por ejemplo, en un encabezado de trama, un encabezado de bloque, un encabezado de segmento, o un encabezado GOP. Los datos de sintaxis de GOP pueden describir una cantidad de tramas en el GOP respectivo, y los datos de sintaxis de trama pueden indicar un modo de codificación/predicción usado para codificar la trama correspondiente.

El codificador 20 de vídeo y el decodificador 30 de vídeo pueden implementarse cada uno como cualquiera de una variedad de circuitos codificadores o decodificadores adecuados, según corresponda, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados de aplicación específica (ASIC), matrices de puertas programables en campo (FPGA), circuitos lógicos discretos, software, hardware, firmware o cualquier combinación de los mismos. Cada uno de los codificadores 20 de vídeo y decodificadores 30 de vídeo puede incluirse en uno o más codificadores o decodificadores, cualquiera de los cuales puede integrarse como parte de un codificador/decodificador de vídeo combinado (CODEC). Un dispositivo que incluye un codificador 20 de vídeo y/o un decodificador 30 de vídeo puede comprender un circuito integrado, un microprocesador y/o un dispositivo de comunicación inalámbrica, tal como un teléfono celular.

La FIG. 2 es un diagrama de bloques que ilustra un ejemplo de codificador 20 de vídeo que puede implementar técnicas para codificar bloques de vídeo usando un proceso de conversión de espacio de color. El codificador 20 de vídeo puede realizar una codificación intra e inter de bloques de vídeo dentro de porciones de vídeo. La codificación intra se basa en la predicción espacial para reducir o eliminar la redundancia espacial en el vídeo dentro de una trama o imagen de vídeo determinado. La codificación inter se basa en la predicción temporal para reducir o eliminar la redundancia temporal en el vídeo dentro de tramas o imágenes adyacentes de una secuencia de vídeo. El modo intra (modo I) puede referirse a cualquiera de varios modos de codificación basados en el espacio. Los modos inter, tales como la predicción unidireccional (modo P) o la bipredicción (modo B), pueden referirse a cualquiera de varios modos de codificación basados en el tiempo.

Como se muestra en la FIG. 2, el codificador 20 de vídeo recibe un bloque de vídeo actual dentro de una trama de vídeo que se va a codificar. En el ejemplo de la FIG. 2, el codificador 20 de vídeo incluye una unidad 40 de selección de modo, una memoria 64 de imagen de referencia, un sumador 50, una unidad 52 de procesamiento de transformación, una unidad 54 de cuantificación y una unidad 56 de codificación de entropía. La unidad 40 de selección de modo, a su vez, incluye la unidad 44 de compensación de movimiento, la unidad 42 de estimación de movimiento, la unidad 46 de predicción intra y la unidad 48 de partición. Para la reconstrucción de bloques de vídeo, el codificador 20 de vídeo también incluye una unidad 58 de cuantificación inversa, una unidad 60 de transformación inversa y un sumador 62. Un filtro de desbloqueo (no mostrado en la FIG. 2) también se puede incluir para filtrar los límites de los bloques y eliminar los artefactos de bloqueo del vídeo reconstruido. Si se desea, el filtro de desbloqueo normalmente filtraría la salida del sumador 62. También se pueden utilizar filtros adicionales (en bucle o post bucle) además del filtro de desbloqueo. Dichos filtros no se muestran por brevedad, pero si se desea, pueden filtrar la salida del sumador 50 (como un filtro en bucle).

Durante el proceso de codificación, el codificador 20 de vídeo recibe una trama o segmento de vídeo que se va a codificar. La trama o segmento se puede dividir en varios bloques de vídeo. La unidad 42 de estimación de movimiento y la unidad 44 de compensación de movimiento realizan codificación predictiva inter del bloque de vídeo recibido con respecto a uno o más bloques en una o más tramas de referencia para proporcionar predicción temporal. La unidad 46 de predicción intra puede realizar alternativamente codificación predictiva intra del bloque de vídeo recibido con respecto a uno o más bloques vecinos en la misma trama o segmento que el bloque que se va a codificar para proporcionar predicción espacial. El codificador 20 de vídeo puede realizar múltiples pases de codificación, por ejemplo, para seleccionar un modo de codificación apropiado para cada bloque de datos de vídeo.

Además, la unidad 48 de partición puede dividir bloques de datos de vídeo en subbloques, basándose en la evaluación de esquemas de partición anteriores en pasadas de codificación anteriores. Por ejemplo, la unidad 48 de partición puede dividir inicialmente una trama o segmento en LCU, y dividir cada una de las LCU en subCU basándose en un análisis de distorsión de velocidad (por ejemplo, optimización de distorsión de velocidad). La unidad 40 de selección de modo puede producir además una estructura de datos de árbol cuádruple indicativa de la partición de una LCU en subCU. Las CU de nodo hoja del árbol cuádruple pueden incluir una o más PU y una o más TU.

La unidad 40 de selección de modo puede seleccionar uno de los modos de codificación, intra o inter, por ejemplo, basándose en resultados de error, y proporciona el bloque codificado intra o inter resultante al sumador 50 para generar datos de bloque residuales y al sumador 62 para reconstruir los bloques codificados para usarlos como trama de referencia. La unidad 40 de selección de modo también proporciona elementos de sintaxis, tales como vectores de

movimiento, indicadores modo intra, información de partición y otra información de sintaxis similar, a la unidad 56 de codificación de entropía.

La unidad 42 de estimación de movimiento y la unidad 44 de compensación de movimiento pueden estar altamente integradas, pero se ilustran por separado con fines conceptuales. La estimación de movimiento, realizada por la unidad 42 de estimación de movimiento, es el proceso de generar vectores de movimiento, que estiman el movimiento para bloques de vídeo. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de una PU de un bloque de vídeo dentro de una trama de vídeo o imagen actual con respecto a un bloque predictivo dentro de una trama de referencia (u otra unidad codificada) en relación con el bloque actual que se está codificando dentro de la trama actual (u otra unidad codificada). Un bloque predictivo es un bloque que coincide estrechamente con el bloque que se va a codificar, en términos de diferencia de píxeles, que puede determinarse mediante la suma de la diferencia absoluta (SAD), la suma de la diferencia cuadrada (SSD) u otras métricas de diferencia. En algunos ejemplos, el codificador 20 de vídeo puede calcular valores para posiciones de píxeles subenteros de imágenes de referencia almacenadas en la memoria 64 de imagen de referencia. Por ejemplo, el codificador 20 de vídeo puede interpolar valores de posiciones de un cuarto de píxel, posiciones de un octavo de píxel u otras posiciones fraccionarias de píxel de una imagen de referencia. Por lo tanto, la unidad 42 de estimación de movimiento puede realizar una búsqueda de movimiento con respecto a las posiciones de píxeles completas y las posiciones de píxeles fraccionales y generar un vector de movimiento con precisión de píxeles fraccionarios.

La unidad 42 de estimación de movimiento calcula un vector de movimiento para una PU de un bloque de vídeo en un segmento codificado inter comparando la posición de la PU con la posición de un bloque predictivo de una imagen de referencia. La imagen de referencia puede seleccionarse de una primera lista de imágenes de referencia (Lista 0) o una segunda lista de imágenes de referencia (Lista 1), cada una de las cuales identifica una o más imágenes de referencia almacenadas en la memoria 64 de imagen de referencia. La unidad 42 de estimación de movimiento envía el vector de movimiento calculado a la unidad 56 de codificación de entropía y a la unidad 44 de compensación de movimiento.

La compensación de movimiento, realizada por la unidad 44 de compensación de movimiento, puede implicar buscar o generar el bloque predictivo basándose en el vector de movimiento determinado por la unidad 42 de estimación de movimiento. Nuevamente, la unidad 42 de estimación de movimiento y la unidad 44 de compensación de movimiento pueden integrarse funcionalmente, en algunos ejemplos. Al recibir el vector de movimiento para la PU del bloque de vídeo actual, la unidad 44 de compensación de movimiento puede localizar el bloque predictivo al que apunta el vector de movimiento en una de las listas de imágenes de referencia. El sumador 50 forma un bloque de vídeo residual restando los valores de píxeles del bloque predictivo de los valores de píxeles del bloque de vídeo actual que se está codificando, formando valores de diferencia de píxeles, como se analiza a continuación. En general, la unidad 42 de estimación de movimiento realiza una estimación de movimiento en relación con los componentes de luma, y la unidad 44 de compensación de movimiento usa vectores de movimiento calculados con base en los componentes de luma tanto para los componentes de croma como para los componentes de luma. La unidad 40 de selección de modo también puede generar elementos de sintaxis asociados con los bloques de vídeo y el segmento de vídeo para su uso por el decodificador 30 de vídeo al decodificar los bloques de vídeo del segmento de vídeo.

La unidad 46 de predicción intra puede predecir intra un bloque actual, como alternativa a la predicción inter realizada por la unidad 42 de estimación de movimiento y la unidad 44 de compensación de movimiento, como se describió anteriormente. En particular, la unidad 46 de predicción intra puede determinar un modo de predicción intra a utilizar para codificar un bloque actual. En algunos ejemplos, la unidad 46 de predicción intra puede codificar un bloque actual usando diversos modos de predicción intra, por ejemplo, durante pases de codificación separados, y la unidad 46 de predicción intra (o la unidad 40 de selección de modo, en algunos ejemplos) puede seleccionar un modo de predicción intra apropiado para usar entre los modos probados.

Por ejemplo, la unidad 46 de predicción intra puede calcular valores de distorsión de velocidad usando un análisis de distorsión de velocidad para los diversos modos de predicción intra probados, y seleccionar el modo de predicción intra que tenga las mejores características de distorsión de velocidad entre los modos probados. El análisis de distorsión de velocidad generalmente determina una cantidad de distorsión (o error) entre un bloque codificado y un bloque original no codificado que se codificó para producir el bloque codificado, así como una velocidad de bits (es decir, una cantidad de bits) utilizada para producir el bloque codificado. La unidad 46 de predicción intra puede calcular relaciones a partir de las distorsiones y velocidades para los diversos bloques codificados para determinar qué modo de predicción intra presenta el mejor valor de distorsión de velocidad para el bloque.

Después de seleccionar un modo de predicción intra para un bloque, la unidad 46 de predicción intra puede proporcionar información indicativa del modo de predicción intra seleccionado para el bloque a la unidad 56 de codificación de entropía. La unidad 56 de codificación de entropía puede codificar la información que indica el modo de predicción intra seleccionado. El codificador 20 de vídeo puede incluir en los datos de configuración del flujo de bits transmitido, que pueden incluir una pluralidad de tablas de índice de modo de predicción intra y una pluralidad de tablas de índice de modo de predicción intra modificadas (también denominadas tablas de mapeo de palabras de código), definiciones de contextos de codificación para diversos bloques e indicaciones de un modo de predicción intra

más probable, una tabla de índice de modo de predicción intra y una tabla de índice de modo de predicción intra modificada para usar en cada uno de los contextos.

El codificador 20 de vídeo forma un bloque de vídeo residual restando los datos de predicción de la unidad 40 de selección de modo del bloque de vídeo original que se está codificando. Sumador 50 representa el componente o componentes que realizan esta operación de resta. La unidad 52 de procesamiento de transformación aplica una transformación, tal como una transformación de coseno discreta (DCT) o una transformación conceptualmente similar, al bloque residual, produciendo un bloque de vídeo que comprende valores de coeficientes de transformación residual. La unidad 52 de procesamiento de transformación puede realizar otras transformaciones que son conceptualmente similares a DCT. También podrían usarse transformaciones de ondícula, transformaciones de números enteros, transformaciones de subbanda u otros tipos de transformaciones. En cualquier caso, la unidad 52 de procesamiento de transformación aplica la transformación al bloque residual, produciendo un bloque de coeficientes de transformación residual. La transformación puede convertir la información residual de un dominio de valor de píxel a un dominio de transformación, tal como un dominio de frecuencia. La unidad 52 de procesamiento de transformación puede enviar los coeficientes de transformación resultantes a la unidad 54 de cuantificación. La unidad 54 de cuantificación cuantifica los coeficientes de transformación para reducir aún más la velocidad de bits. El proceso de cuantificación puede reducir la profundidad de bits asociada con algunos o todos los coeficientes. El grado de cuantificación se puede modificar ajustando un parámetro de cuantificación. En algunos ejemplos, la unidad 54 de cuantificación puede realizar entonces una exploración de la matriz que incluye los coeficientes de transformación cuantificados. Alternativamente, la unidad 56 de codificación de entropía puede realizar la exploración.

Después de la cuantificación, la unidad 56 de codificación de entropía codifica los coeficientes de transformación cuantificados. Por ejemplo, la unidad 56 de codificación de entropía puede realizar codificación de longitud variable adaptativa al contexto (CAVLC), codificación aritmética binaria adaptativa al contexto (CABAC), codificación aritmética binaria adaptativa al contexto (SBAC) basada en sintaxis, codificación de entropía de partición de intervalos de probabilidad (PIPE) u otra técnica de codificación de entropía. En el caso de la codificación de entropía basada en contexto, el contexto puede basarse en bloques vecinos. Después de la codificación de entropía por la unidad 56 de codificación de entropía, el flujo de bits codificado puede transmitirse a otro dispositivo (por ejemplo, el decodificador 30 de vídeo) o archivar para su posterior transmisión o recuperación.

De acuerdo con las técnicas de esta divulgación, la unidad 56 de codificación de entropía del codificador 20 de vídeo puede realizar una o más técnicas de la presente divulgación. Por ejemplo, la unidad 56 de codificación de entropía del codificador 20 de vídeo puede determinar una profundidad de bits de un componente luma de los datos de vídeo y una profundidad de bits de un componente croma de los datos de vídeo. La unidad 56 de codificación de entropía puede además, en respuesta a que la profundidad de bits del componente luma sea diferente de la profundidad de bits del componente croma, modificar una o ambas de la profundidad de bits del componente luma y la profundidad de bits del componente croma tal que las profundidades de bits sean iguales. La unidad 56 de codificación de entropía del codificador 20 de vídeo puede aplicar además el proceso de transformación del espacio de color a los datos de vídeo modificados.

La unidad 58 de cuantificación inversa y la unidad 60 de transformación inversa aplican cuantificación inversa y transformación inversa, respectivamente, para reconstruir el bloque residual en el dominio de píxeles, por ejemplo, para uso posterior como bloque de referencia. La unidad 44 de compensación de movimiento puede calcular un bloque de referencia sumando el bloque residual a un bloque predictivo de una de las tramas de la memoria 64 de imágenes de referencia. La unidad 44 de compensación de movimiento también puede aplicar uno o más filtros de interpolación al bloque residual reconstruido para calcular valores de píxeles subenteros para su uso en la estimación de movimiento. El sumador 62 añade el bloque residual reconstruido al bloque de predicción con compensación de movimiento producido por la unidad 44 de compensación de movimiento para producir un bloque de vídeo reconstruido para su almacenamiento en la memoria 64 de imágenes de referencia. El bloque de vídeo reconstruido puede ser utilizado por la unidad 42 de estimación de movimiento y la unidad 44 de compensación de movimiento como bloque de referencia para codificar inter un bloque en una X de vídeo posterior.

De esta manera, la unidad 56 de codificación de entropía del codificador 20 de vídeo puede realizar una o más técnicas de la presente divulgación. Por ejemplo, de acuerdo con las técnicas de esta divulgación, la unidad 56 de codificación de entropía del codificador 20 de vídeo puede realizar una o más técnicas de la presente divulgación. Por ejemplo, la unidad 56 de codificación de entropía del codificador 20 de vídeo puede determinar una profundidad de bits de un componente luma de los datos de vídeo y una profundidad de bits de un componente croma de los datos de vídeo. La unidad 56 de codificación de entropía puede además, en respuesta a que la profundidad de bits del componente luma sea diferente de la profundidad de bits del componente croma, modificar una o ambas de la profundidad de bits del componente luma y la profundidad de bits del componente croma tal que las profundidades de bits sean iguales. La unidad 56 de codificación de entropía del codificador 20 de vídeo puede aplicar además el proceso de transformación del espacio de color a los datos de vídeo modificados.

En algunos ejemplos, los datos de vídeo pueden ser un primer conjunto de datos de vídeo. Como tal, para un segundo conjunto de datos de vídeo, la unidad 56 de codificación de entropía del codificador 20 de vídeo puede determinar una profundidad de bits de un componente luma del segundo conjunto de datos de vídeo y una profundidad de bits de un

componente croma del segundo conjunto de datos de vídeo. En respuesta a que la profundidad de bits del componente luma del segundo conjunto de datos de vídeo sea diferente de la profundidad de bits del componente croma del segundo conjunto de datos de vídeo, la unidad 56 de codificación de entropía del codificador 20 de vídeo puede desactivar la transformación de color para el segundo conjunto de datos de vídeo. En algunos de tales ejemplos, la

En algunos ejemplos, cuando la unidad 56 de codificación de entropía del codificador 20 de vídeo modifica una o ambas de la profundidad de bits del componente luma y la profundidad de bits del componente croma, la unidad 56 de codificación de entropía del codificador 20 de vídeo puede realizar una operación de desplazamiento bit a bit en el datos de vídeo de uno o ambos del componente luma y del componente croma antes de aplicar el proceso de transformación del espacio de color. Una operación de desplazamiento bit a bit opera en uno o más patrones de bits o números binarios al nivel de sus bits individuales, con ceros reemplazando los bits descartados.

En algunos ejemplos, la unidad 56 de codificación de entropía del codificador 20 de vídeo puede determinar además usar conversión de espacio de color para codificar los datos de vídeo. Al hacerlo, la unidad 56 de codificación de entropía del codificador 20 de vídeo puede establecer un valor de un elemento de sintaxis de los datos de vídeo para indicar el uso de la conversión del espacio de color en respuesta a la determinación de usar la conversión del espacio de color para la unidad de codificación. En algunos ejemplos, el elemento de sintaxis comprende un indicador de un bit. En algunos ejemplos, el elemento de sintaxis se señala cuando la unidad de codificación se codifica usando un modo distinto del modo de modulación de código de pulso intra (IPCM). En otros ejemplos, el elemento de sintaxis se señala sólo cuando hay coeficientes distintos de cero en una unidad de transformación de la unidad de codificación. En algunos ejemplos, un valor de 1 para el elemento de sintaxis indica que la unidad de codificación se codificó mediante conversión de espacio de color.

En algunos ejemplos, es posible que el elemento de sintaxis no esté señalado. Por ejemplo, el elemento de sintaxis puede no señalizarse cuando la unidad de codificación está codificada intra y cuando un modo de predicción de luma y un modo de predicción de croma de una unidad de predicción dentro de la unidad de codificación son diferentes. En otro ejemplo, es posible que el elemento de sintaxis no se señalice cuando la unidad de codificación se codifica con un modo de paleta.

En algunos ejemplos, el elemento de sintaxis es un primer elemento de sintaxis y los datos de vídeo son un primer conjunto de datos de vídeo. En tales ejemplos, la unidad 56 de codificación de entropía del codificador 20 de vídeo puede determinar un valor de un índice de paleta para un primer píxel en una segunda unidad de codificación de un segundo conjunto de datos de vídeo. La unidad 56 de codificación de entropía del codificador 20 de vídeo puede determinar además valores de índices de paleta para uno o más píxeles siguientes en un orden de exploración inmediatamente posterior al primer píxel. Determinar los valores de los índices de paleta puede comprender determinar los valores de los índices de paleta hasta que un píxel tenga un índice de paleta con un valor diferente al valor del índice de paleta para el primer píxel. La unidad 56 de codificación de entropía del codificador 20 de vídeo puede determinar además un número de valores de índice de paleta determinados para los siguientes píxeles. En función del número de valores de índice de paleta, la unidad 56 de codificación de entropía del codificador 20 de vídeo puede codificar el índice de paleta usando un primer modo o un segundo modo. Cuando el número de valores de índice de paleta es mayor que uno, la unidad 56 de codificación de entropía del codificador 20 de vídeo puede establecer un valor de un segundo elemento de sintaxis para indicar que el primer modo se usó para codificar el índice de paleta y establecer un valor de un tercer elemento de sintaxis igual al número de valores del índice de la paleta. Cuando el número de valores de índice de paleta es menor o igual a uno, la unidad 56 de codificación de entropía del codificador 20 de vídeo puede establecer el valor del segundo elemento de sintaxis para indicar que se usó el segundo modo para codificar el índice de paleta. En algunos ejemplos, el primer modo es un modo de ejecución y el segundo modo es un modo de píxel.

En algunos ejemplos, la codificación del elemento de sintaxis puede comprender la unidad 56 de codificación de entropía del codificador 20 de vídeo que codifica datos de un mensaje de información de mejora suplementaria (SEI) que indica si la unidad de codificación se codificó usando conversión de espacio de color.

La FIG. 3 es un diagrama de bloques que ilustra un ejemplo de decodificador 30 de vídeo que puede implementar técnicas para decodificar bloques de vídeo, algunos de los cuales se codificaron usando un proceso de conversión de espacio de color. En el ejemplo de la FIG. 3, el decodificador 30 de vídeo incluye una unidad 70 de decodificación de entropía, una unidad 72 de compensación de movimiento, una unidad 74 de predicción intra, una unidad 76 de cuantificación inversa, una unidad 78 de transformación inversa, una memoria 82 de imagen de referencia y un sumador 80. El decodificador 30 de vídeo puede, en algunos ejemplos, realizar un paso de decodificación generalmente recíproco al paso de codificación descrito con respecto al codificador 20 de vídeo (FIG. 2). La unidad 72 de compensación de movimiento puede generar datos de predicción basados en vectores de movimiento recibidos desde la unidad 70 de decodificación de entropía, mientras que la unidad 74 de predicción intra puede generar datos de predicción basados en indicadores de modo de predicción intra recibidos de la unidad 70 de decodificación de entropía.

Durante el proceso de decodificación, el decodificador 30 de vídeo recibe un flujo de bits de vídeo codificado que representa bloques de vídeo de un segmento de vídeo codificado y elementos de sintaxis asociados del codificador 20 de vídeo. La unidad 70 de decodificación de entropía del decodificador 30 de vídeo decodifica por entropía el flujo de bits para generar coeficientes cuantificados, vectores de movimiento o indicadores de modo de predicción intra y otros elementos de sintaxis. La unidad 70 de decodificación de entropía envía los vectores de movimiento y otros elementos de sintaxis a la unidad 72 de compensación de movimiento. El decodificador 30 de vídeo puede recibir los elementos de sintaxis en el nivel de segmento de vídeo y/o en el nivel de bloque de vídeo.

De acuerdo con las técnicas de esta divulgación, la unidad 70 de decodificación de entropía del decodificador 30 de vídeo puede realizar una o más técnicas de la presente divulgación. Por ejemplo, la unidad 70 de decodificación de entropía del codificador 30 de vídeo puede determinar una profundidad de bits de un componente luma de los datos de vídeo y una profundidad de bits de un componente croma de los datos de vídeo. La unidad 70 de decodificación de entropía del codificador 30 de vídeo puede además, en respuesta a que la profundidad de bits del componente luma sea diferente de la profundidad de bits del componente croma, modificar una o ambas de la profundidad de bits del componente luma y la profundidad de bits del componente croma de modo que las profundidades de bits sean iguales. La unidad 70 de decodificación de entropía del codificador 30 de vídeo puede aplicar además el proceso de transformación inversa del espacio de color a los datos de vídeo modificados.

Cuando el segmento de vídeo se codifica como un segmento codificado intra (I), la unidad 74 de predicción intra puede generar datos de predicción para un bloque de vídeo del segmento de vídeo actual basándose en un modo de predicción intra señalado y datos de bloques previamente decodificados de la trama o imagen actual. Cuando la trama de vídeo se codifica como un segmento codificado inter (es decir, B, P o GPB), la unidad 72 de compensación de movimiento produce bloques predictivos para un bloque de vídeo del segmento de vídeo actual basándose en los vectores de movimiento y otros elementos de sintaxis recibidos de la unidad 70 de decodificación de entropía. Los bloques predictivos pueden producirse a partir de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. El decodificador 30 de vídeo puede construir las listas de tramas de referencia, Lista 0 y Lista 1, usando técnicas de construcción predeterminadas basadas en imágenes de referencia almacenadas en la memoria 82 de imágenes de referencia. La unidad 72 de compensación de movimiento determina la información de predicción para un bloque de vídeo del segmento de vídeo actual analizando los vectores de movimiento y otros elementos de sintaxis, y utiliza la información de predicción para producir los bloques predictivos para el bloque de vídeo actual que se está decodificando. Por ejemplo, la unidad 72 de compensación de movimiento usa algunos de los elementos de sintaxis recibidos para determinar un modo de predicción (por ejemplo, predicción intra o inter) usado para codificar los bloques de vídeo del segmento de vídeo, un tipo de segmento de predicción inter (por ejemplo, B segmento, segmento P o segmento GPB), información de construcción para una o más de las listas de imágenes de referencia para el segmento, vectores de movimiento para cada bloque de vídeo codificado inter del segmento, estado de predicción inter para cada bloque de vídeo codificado inter del segmento y otra información para decodificar los bloques de vídeo en el segmento de vídeo actual.

La unidad 72 de compensación de movimiento también puede realizar una interpolación basada en filtros de interpolación. La unidad 72 de compensación de movimiento puede usar filtros de interpolación como los que usa el codificador 20 de vídeo durante la codificación de los bloques de vídeo para calcular valores interpolados para píxeles subenteros de bloques de referencia. En este caso, la unidad 72 de compensación de movimiento puede determinar los filtros de interpolación utilizados por el codificador 20 de vídeo a partir de los elementos de sintaxis recibidos y utilizar los filtros de interpolación para producir bloques predictivos.

La unidad 76 de cuantificación inversa cuantifica inversamente, es decir, descuantifica, los coeficientes de transformación cuantificados proporcionados en el flujo de bits y decodificados por la unidad 70 de decodificación de entropía. El proceso de cuantificación inversa puede incluir el uso de un parámetro de cuantificación QP_Y calculado por el decodificador 30 de vídeo para cada bloque de vídeo en el segmento de vídeo para determinar un grado de cuantificación y, asimismo, un grado de cuantificación inversa que debería aplicarse.

La unidad 78 de transformación inversa aplica una transformación inversa, por ejemplo, una DCT inversa, una transformación de enteros inversa o un proceso de transformación inversa conceptualmente similar, a los coeficientes de transformación para producir bloques residuales en el dominio de píxeles.

Después de que la unidad 72 de compensación de movimiento genera el bloque predictivo para el bloque de vídeo actual basándose en los vectores de movimiento y otros elementos de sintaxis, el decodificador 30 de vídeo forma un bloque de vídeo decodificado sumando los bloques residuales de la unidad 78 de transformación inversa con los correspondientes bloques predictivos generados por la unidad 72 de compensación de movimiento. Sumador 80 representa el componente o componentes que realizan esta operación de suma. Si se desea, también se puede aplicar un filtro de desbloqueo para filtrar los bloques decodificados con el fin de eliminar artefactos de bloqueo. También se pueden usar otros filtros de bucle (ya sea en el bucle de codificación o después del bucle de codificación) para suavizar las transiciones de píxeles o mejorar de otro modo la calidad del vídeo. Los bloques de vídeo decodificados en una trama o imagen determinados se almacenan luego en la memoria 82 de imágenes de referencia, que almacena imágenes de referencia utilizadas para la compensación de movimiento posterior. La memoria 82 de imágenes de

referencia también almacena vídeo decodificado para su posterior presentación en un dispositivo de visualización, tal como el dispositivo 32 de visualización de la FIG. 1.

De esta manera, la unidad 70 de decodificación de entropía del codificador 30 de vídeo puede realizar una o más técnicas de la presente divulgación. Por ejemplo, la unidad 70 de decodificación de entropía del codificador 30 de vídeo puede determinar una profundidad de bits de un componente luma de los datos de vídeo y una profundidad de bits de un componente croma de los datos de vídeo. La unidad 70 de decodificación de entropía del codificador 30 de vídeo puede además, en respuesta a que la profundidad de bits del componente luma sea diferente de la profundidad de bits del componente croma, modificar una o ambas de la profundidad de bits del componente luma y la profundidad de bits del componente croma de modo que las profundidades de bits sean iguales. La unidad 70 de decodificación de entropía del codificador 30 de vídeo puede aplicar además el proceso de transformación inversa del espacio de color a los datos de vídeo modificados.

En algunos ejemplos, los datos de vídeo pueden ser un primer conjunto de datos de vídeo. Como tal, para un segundo conjunto de datos de vídeo, la unidad 70 de decodificación de entropía del codificador 30 de vídeo puede determinar una profundidad de bits de un componente luma del segundo conjunto de datos de vídeo y una profundidad de bits de un componente croma del segundo conjunto de datos de vídeo. En respuesta a que la profundidad de bits del componente luma del segundo conjunto de datos de vídeo sea diferente de la profundidad de bits del componente croma del segundo conjunto de datos de vídeo, la unidad 70 de decodificación de entropía del codificador 30 de vídeo puede desactivar la transformación de color para el segundo conjunto de datos de vídeo. En algunos de tales ejemplos, la unidad 70 de decodificación de entropía del codificador 30 de vídeo puede establecer además un segundo elemento de sintaxis de un conjunto de parámetros de imagen del segundo conjunto de datos de vídeo que indica que el proceso de transformación del espacio de color está deshabilitado para el segundo conjunto de datos de vídeo.

En algunos ejemplos, cuando la unidad 70 de decodificación de entropía del codificador 30 de vídeo modifica una o ambas de la profundidad de bits del componente luma y la profundidad de bits del componente croma, la unidad 70 de decodificación de entropía del codificador 30 de vídeo puede realizar una operación de desplazamiento bit a bit en el datos de vídeo de uno o ambos del componente luma y del componente croma antes de aplicar el proceso de transformación del espacio de color. Una operación de desplazamiento bit a bit opera en uno o más patrones de bits o números binarios al nivel de sus bits individuales, con ceros reemplazando los bits descartados.

En algunos ejemplos, la unidad 70 de decodificación de entropía del decodificador 30 de vídeo puede decodificar un elemento de sintaxis de una unidad de codificación de datos de vídeo. El elemento de sintaxis puede indicar si la unidad de codificación se codificó mediante conversión de espacio de color. En algunos ejemplos, el elemento de sintaxis puede comprender un indicador de un bit. En algunos ejemplos, la unidad de codificación se codifica en un modo distinto del modo de modulación de código pulso intra (IPCM) y el elemento de sintaxis se señala sólo para unidades de codificación que utilizan un modo distinto del modo IPCM. En algunos ejemplos, el elemento de sintaxis indica que la unidad de codificación se codificó usando conversión de espacio de color cuando hay coeficientes distintos de cero en una unidad de transformación de la unidad de codificación. En algunos ejemplos, un valor de 1 para el elemento de sintaxis indica que la unidad de codificación se codificó mediante conversión de espacio de color. La unidad 70 de decodificación de entropía del decodificador 30 de vídeo puede determinar además si un valor del elemento de sintaxis indica que la unidad de codificación se codificó usando conversión de espacio de color. En respuesta a la determinación de que el elemento de sintaxis indica que la unidad de codificación se codificó usando conversión de espacio de color, la unidad 70 de decodificación de entropía del decodificador 30 de vídeo puede aplicar el proceso de transformación inversa de espacio de color.

En algunos ejemplos, el elemento de sintaxis puede indicar que no se utilizó la conversión del espacio de color para codificar la unidad de codificación. Por ejemplo, el elemento de sintaxis puede indicar que no se usó conversión de espacio de color para codificar la unidad de codificación cuando la unidad de codificación está codificada intra y cuando un modo de predicción de luma y un modo de predicción de croma de una unidad de predicción dentro de la unidad de codificación son diferentes. En otro ejemplo, el elemento de sintaxis puede indicar que no se usó la conversión del espacio de color para codificar la unidad de codificación cuando la unidad de codificación está codificada con un modo de paleta. En estos ejemplos, el flujo de bits recibido puede no incluir el elemento de sintaxis.

En algunos ejemplos, el elemento de sintaxis es un primer elemento de sintaxis y los datos de vídeo son un primer conjunto de datos de vídeo. En tales ejemplos, la unidad 70 de decodificación de entropía del decodificador 30 de vídeo puede determinar además un valor de un segundo elemento de sintaxis de un segundo conjunto de datos de vídeo. El segundo elemento de sintaxis puede indicar si se usó un modo de paleta para codificar el segundo conjunto de datos de vídeo. En respuesta al segundo elemento de sintaxis que indica que se usó el modo de paleta para codificar los datos de vídeo, la unidad 70 de decodificación de entropía del decodificador 30 de vídeo puede determinar un valor de un tercer elemento de sintaxis. El tercer elemento de sintaxis puede indicar si se usa un primer modo o un segundo modo para decodificar un índice de paleta para un píxel en la unidad de codificación. En base al valor determinado del segundo elemento de sintaxis, la unidad 70 de decodificación de entropía del decodificador 30 de vídeo puede decodificar el índice de paleta usando el primer modo o el segundo modo. Cuando se utiliza el primer modo, la unidad 70 de decodificación de entropía del decodificador 30 de vídeo puede determinar un valor del índice de paleta, determinar un valor de un cuarto elemento de sintaxis que indica un número de píxeles en un orden de

exploración inmediatamente posterior al píxel que se está decodificando actualmente, y duplique el resultado de determinar el valor del índice de paleta para los siguientes N píxeles en el orden de exploración, siendo N igual al valor del cuarto elemento de sintaxis. En algunos ejemplos, el primer modo es un modo de ejecución. Cuando se utiliza el segundo modo, la unidad 70 de decodificación de entropía del decodificador 30 de vídeo puede determinar el valor del índice de paleta y generar un valor de muestra de píxel para el píxel, donde el valor de muestra de píxel es igual al valor del índice de paleta. En algunos ejemplos, el segundo modo es un modo de píxel.

En algunos ejemplos, decodificar el elemento de sintaxis comprende que la unidad 70 de decodificación de entropía del decodificador 30 de vídeo esté configurada para decodificar datos de un mensaje de información de mejora suplementaria (SEI) que indica si la unidad de codificación se codificó usando conversión de espacio de color.

La FIG. 4 es un diagrama conceptual que ilustra los 35 modos de predicción HEVC de acuerdo con una o más técnicas de la presente divulgación. En el HEVC actual, para el componente luma de cada Unidad de Predicción (PU), se utiliza un método de predicción intra con 33 modos de predicción angular (indexados de 2 a 34), modo DC (indexado con 1) y modo Planar (indexado con 0), como se describe con respecto a la FIG. 4.

Además de los 35 modos intra anteriores, HEVC también emplea un modo más, denominado 'I-PCM'. En el modo I-PCM, la predicción, la transformación, la cuantificación y la codificación de entropía se omiten mientras que las muestras de predicción se codifican mediante un número predefinido de bits. El objetivo principal del modo I-PCM es manejar la situación en la que otros modos no pueden codificar eficientemente la señal.

La FIG. 5 es un diagrama conceptual que ilustra candidatos de vectores de movimiento vecinos espaciales para modos de fusión y predicción avanzada de vectores de movimiento (AMVP) de acuerdo con una o más técnicas de la presente divulgación. Como se describe con respecto a la FIG. 5, los candidatos de MV espaciales se derivan de los bloques vecinos mostrados en la FIG. 5 para una PU específica (PU0), aunque los métodos que generan los candidatos a partir de los bloques difieren para los modos de fusión y AMVP.

En el modo de fusión, se pueden derivar hasta cuatro candidatos MV espaciales con los órdenes mostrados en la FIG. 5(a) con números, y el orden es el siguiente: izquierda (0), arriba (1), arriba derecha (2), abajo izquierda (3) y arriba izquierda (4), como se muestra en la FIG. 5(a).

En el modo AMVP, los bloques vecinos se dividen en dos grupos: el grupo izquierdo que consta de los bloques 0 y 1, y el grupo superior que consta de los bloques 2, 3 y 4, como se muestra en la FIG. 5(b). Para cada grupo, el candidato potencial en un bloque vecino que hace referencia a la misma imagen de referencia que la indicada por el índice de referencia señalado tiene la prioridad más alta para ser elegido para formar un candidato final del grupo. Es posible que todos los bloques vecinos no contengan un vector de movimiento que apunte a la misma imagen de referencia. Por lo tanto, si no se puede encontrar dicho candidato, el primer candidato disponible se escalará para formar el candidato final, de modo que se puedan compensar las diferencias de distancia temporal.

La FIG. 6 es un diagrama conceptual que ilustra un ejemplo de copia bloque intra (BC) de acuerdo con una o más técnicas de la presente divulgación. Como se describe con respecto a la FIG. 6, la copia bloque intra (BC) se ha incluido en RExt. Se muestra un ejemplo de Intra BC como en la FIG. 6, en el que la CU actual se predice a partir de un bloque ya decodificado de la imagen/segmento actual. El tamaño de bloque Intra BC actual puede ser tan grande como el tamaño de una CU, que oscila entre 8x8 y 64x64, aunque en algunas aplicaciones pueden aplicarse restricciones adicionales.

La FIG. 7 es un diagrama conceptual que ilustra un ejemplo de un bloque objetivo y una muestra de referencia para un bloque intra 8x8, de acuerdo con una o más técnicas de la presente divulgación. Como se describe a continuación con respecto a la FIG. 7, se deriva una matriz de transformación a partir de los valores de muestra de referencia. Se utilizan diferentes muestras de referencia para el caso intra y caso inter. Para el caso de bloque intra, el bloque objetivo y las muestras de referencia se muestran en la FIG. 7. En esta figura, el bloque objetivo consta de muestras rayadas y las referencias son muestras rayadas y punteadas.

Para el caso de bloque inter, las muestras de referencia para la derivación de la matriz son las mismas que para la compensación de movimiento. Para realizar la operación de cambio, las muestras de referencia en el bloque AMP se submuestrean de modo que el número de muestras se convierta en una potencia de dos. Por ejemplo, el número de muestras de referencia en un bloque de 12x16 se reduce a 2/3.

De acuerdo con algunas técnicas de la presente divulgación, se puede aplicar el proceso de transformación del espacio de color. En tales ejemplos, no es necesario indicar si se invoca o no el proceso de transformación. Además, tanto el lado del codificador como el del decodificador pueden derivar la matriz de transformación con el mismo método para evitar la sobrecarga de señalización de la matriz de transformación.

La FIG. 8 es un diagrama de flujo que ilustra un método de ejemplo para codificar un bloque actual. El bloque actual puede comprender una CU actual o una parte de la CU actual. Aunque se describe con respecto al codificador 20 de

vídeo (FIGS. 1 y 2), debe entenderse que se pueden configurar otros dispositivos para realizar un método similar al de la FIG. 8.

En este ejemplo, el codificador 20 de vídeo predice inicialmente el bloque (150) actual. Por ejemplo, el codificador 20 de vídeo puede calcular una o más unidades de predicción (PU) para el bloque actual. El codificador 20 de vídeo puede entonces calcular un bloque residual para el bloque actual, por ejemplo, para producir una unidad de transformación (TU) (152). Para calcular el bloque residual, el codificador 20 de vídeo puede calcular una diferencia entre el bloque original no codificado y el bloque previsto para el bloque actual. La unidad 56 de codificación de entropía del codificador 20 de vídeo puede determinar una profundidad de bits de un componente luma de los datos de vídeo y una profundidad de bits de un componente croma de los datos (154) de vídeo. La unidad 56 de codificación de entropía puede además, en respuesta a que la profundidad de bits del componente luma sea diferente de la profundidad de bits del componente croma, modificar una o ambas de la profundidad de bits del componente luma y la profundidad de bits del componente croma tal que las profundidades de bits sean iguales (156). La unidad 56 de codificación de entropía del codificador 20 de vídeo puede aplicar además el proceso de transformación del espacio de color a los datos (158) de vídeo modificados. El codificador 20 de vídeo puede entonces transformar y cuantificar coeficientes del bloque (160) residual. A continuación, el codificador 20 de vídeo puede explorar los coeficientes de transformación cuantificados del bloque (162) residual. Durante la exploración, o después de la exploración, el codificador 20 de vídeo puede codificar por entropía los coeficientes (164). Por ejemplo, el codificador 20 de vídeo puede codificar los coeficientes usando CAVLC o CABAC. El codificador 20 de vídeo puede entonces generar los datos codificados por entropía del bloque (166). La unidad 56 de codificación de entropía del codificador 20 de vídeo puede aplicar además un proceso de transformación inversa del espacio de color en un bucle de decodificación del proceso (168) de codificación. En las técnicas de esta divulgación, el proceso de transformación inversa del espacio de color es independiente de si el modo de codificación es el modo de codificación con pérdidas o el modo de codificación sin pérdidas.

En algunos ejemplos, los datos de vídeo pueden ser un primer conjunto de datos de vídeo. Como tal, para un segundo conjunto de datos de vídeo, la unidad 56 de codificación de entropía del codificador 20 de vídeo puede determinar una profundidad de bits de un componente luma del segundo conjunto de datos de vídeo y una profundidad de bits de un componente croma del segundo conjunto de datos de vídeo. En respuesta a que la profundidad de bits del componente luma del segundo conjunto de datos de vídeo sea diferente de la profundidad de bits del componente croma del segundo conjunto de datos de vídeo, la unidad 56 de codificación de entropía del codificador 20 de vídeo puede desactivar la transformación de color para el segundo conjunto de datos de vídeo. En algunos de tales ejemplos, la unidad 56 de codificación de entropía del codificador 20 de vídeo puede establecer además un segundo elemento de sintaxis de un conjunto de parámetros de imagen del segundo conjunto de datos de vídeo que indica que el proceso de transformación del espacio de color está desactivado para el segundo conjunto de datos de vídeo.

En algunos ejemplos, cuando la unidad 56 de codificación de entropía del codificador 20 de vídeo modifica una o ambas de la profundidad de bits del componente luma y la profundidad de bits del componente croma, la unidad 56 de codificación de entropía del codificador 20 de vídeo puede realizar una operación de desplazamiento bit a bit en los datos de vídeo de uno o ambos del componente luma y del componente croma antes de aplicar el proceso de transformación del espacio de color. Una operación de desplazamiento bit a bit opera en uno o más patrones de bits o números binarios al nivel de sus bits individuales, con ceros reemplazando los bits descartados.

En algunos ejemplos, la unidad 56 de codificación de entropía del codificador 20 de vídeo puede determinar además usar conversión de espacio de color para codificar los datos de vídeo. Al hacerlo, la unidad 56 de codificación de entropía del codificador 20 de vídeo puede establecer un valor de un elemento de sintaxis de los datos de vídeo para indicar el uso de la conversión del espacio de color en respuesta a la determinación de usar la conversión del espacio de color para la unidad de codificación. En algunos ejemplos, el elemento de sintaxis comprende un indicador de un bit. En algunos ejemplos, el elemento de sintaxis se señala cuando la unidad de codificación se codifica usando un modo distinto del modo de modulación de código de pulso intra (IPCM). En otros ejemplos, el elemento de sintaxis se señala sólo cuando hay coeficientes distintos de cero en una unidad de transformación de la unidad de codificación. En algunos ejemplos, un valor de 1 para el elemento de sintaxis indica que la unidad de codificación se codificó mediante conversión de espacio de color.

En algunos ejemplos, es posible que el elemento de sintaxis no esté señalado. Por ejemplo, el elemento de sintaxis puede no señalizarse cuando la unidad de codificación está codificada intra y cuando un modo de predicción de luma y un modo de predicción de croma de una unidad de predicción dentro de la unidad de codificación son diferentes. En otro ejemplo, es posible que el elemento de sintaxis no se señale cuando la unidad de codificación se codifica con un modo de paleta. En estos ejemplos, el elemento de sintaxis puede no estar presente en un flujo de bits recibido que comprende los datos de vídeo, y decodificar el elemento de sintaxis puede comprender inferir el valor del elemento de sintaxis.

En algunos ejemplos, el elemento de sintaxis es un primer elemento de sintaxis y los datos de vídeo son un primer conjunto de datos de vídeo. En tales ejemplos, la unidad 56 de codificación de entropía del codificador 20 de vídeo puede determinar un valor de un índice de paleta para un primer píxel en una segunda unidad de codificación de un segundo conjunto de datos de vídeo. La unidad 56 de codificación de entropía del codificador 20 de vídeo puede

determinar además valores de índices de paleta para uno o más píxeles siguientes en un orden de exploración inmediatamente posterior al primer píxel. Determinar los valores de los índices de paleta puede comprender determinar los valores de los índices de paleta hasta que un píxel tenga un índice de paleta con un valor diferente al valor del índice de paleta para el primer píxel. La unidad 56 de codificación de entropía del codificador 20 de vídeo puede determinar además un número de valores de índice de paleta determinados para los siguientes píxeles. En función del número de valores de índice de paleta, la unidad 56 de codificación de entropía del codificador 20 de vídeo puede codificar el índice de paleta usando un primer modo o un segundo modo. Cuando el número de valores de índice de paleta es mayor que uno, la unidad 56 de codificación de entropía del codificador 20 de vídeo puede establecer un valor de un segundo elemento de sintaxis para indicar que el primer modo se usó para codificar el índice de paleta y establecer un valor de un tercer elemento de sintaxis igual al número de valores del índice de la paleta. Cuando el número de valores de índice de paleta es menor o igual a uno, la unidad 56 de codificación de entropía del codificador 20 de vídeo puede establecer el valor del segundo elemento de sintaxis para indicar que se usó el segundo modo para codificar el índice de paleta. En algunos ejemplos, el primer modo es un modo de ejecución y el segundo modo es un modo de píxel.

En algunos ejemplos, la codificación del elemento de sintaxis puede comprender la unidad 56 de codificación entropía del codificador 20 de vídeo que codifica datos de un mensaje de información de mejora suplementaria (SEI) que indica si la unidad de codificación se codificó usando conversión de espacio de color.

La FIG. 9 es un diagrama de flujo que ilustra un método de ejemplo para decodificar un bloque actual de datos de vídeo. El bloque actual puede comprender una CU actual o una parte de la CU actual. Aunque se describe con respecto al decodificador 30 de vídeo (FIGS. 1 y 3), debe entenderse que se pueden configurar otros dispositivos para realizar un método similar al de la FIG. 9.

El decodificador 30 de vídeo puede predecir el bloque (200) actual, por ejemplo, usando un modo de predicción intra o inter para calcular un bloque predicho para el bloque actual. El decodificador 30 de vídeo también puede recibir datos codificados por entropía para el bloque actual, tales como datos codificados por entropía para coeficientes de un bloque residual correspondiente al bloque (202) actual. El decodificador 30 de vídeo puede decodificar por entropía los datos codificados por entropía para reproducir coeficientes del bloque (204) residual. El decodificador 30 de vídeo puede entonces realizar una exploración inversa de los coeficientes (206) reproducidos para crear un bloque de coeficientes de transformación cuantificados. El decodificador 30 de vídeo puede entonces transformar inversamente y cuantificar inversamente los coeficientes (208). La unidad 70 de decodificación de entropía del decodificador 30 de vídeo puede determinar una profundidad de bits de un componente luma de los datos de vídeo y una profundidad de bits de un componente croma de los datos (210) de vídeo. La unidad 70 de decodificación de entropía del decodificador 30 de vídeo puede además, en respuesta a que la profundidad de bits del componente luma sea diferente de la profundidad de bits del componente croma, modificar una o ambas de la profundidad de bits del componente luma y la profundidad de bits del componente croma de manera que las profundidades de bits sean iguales (212). La unidad 70 de decodificación de entropía del codificador 30 de vídeo también puede aplicar el proceso de transformación inversa del espacio de color a los datos (214) de vídeo modificados. El decodificador 30 de vídeo puede en última instancia decodificar el bloque actual combinando el bloque predicho y el bloque (216) residual.

En algunos ejemplos, los datos de vídeo pueden ser un primer conjunto de datos de vídeo. Como tal, para un segundo conjunto de datos de vídeo, la unidad 70 de decodificación de entropía del codificador 30 de vídeo puede determinar una profundidad de bits de un componente luma del segundo conjunto de datos de vídeo y una profundidad de bits de un componente croma del segundo conjunto de datos de vídeo. En respuesta a que la profundidad de bits del componente luma del segundo conjunto de datos de vídeo sea diferente de la profundidad de bits del componente croma del segundo conjunto de datos de vídeo, la unidad 70 de decodificación de entropía del codificador 30 de vídeo puede desactivar la transformación de color para el segundo conjunto de datos de vídeo. En algunos de tales ejemplos, la unidad 70 de decodificación de entropía del codificador 30 de vídeo puede establecer además un segundo elemento de sintaxis de un conjunto de parámetros de imagen del segundo conjunto de datos de vídeo que indica que el proceso de transformación del espacio de color está deshabilitado para el segundo conjunto de datos de vídeo.

En algunos ejemplos, cuando la unidad 70 de decodificación de entropía del codificador 30 de vídeo modifica una o ambas de la profundidad de bits del componente luma y la profundidad de bits del componente croma, la unidad 70 de decodificación de entropía del codificador 30 de vídeo puede realizar una operación de desplazamiento bit a bit en el datos de vídeo de uno o ambos del componente luma y del componente croma antes de aplicar el proceso de transformación del espacio de color. Una operación de desplazamiento bit a bit opera en uno o más patrones de bits o números binarios al nivel de sus bits individuales, con ceros reemplazando los bits descartados.

En algunos ejemplos, la unidad 70 de decodificación de entropía del decodificador 30 de vídeo puede decodificar un elemento de sintaxis de una unidad de codificación de datos de vídeo. El elemento de sintaxis puede indicar si la unidad de codificación se codificó mediante conversión de espacio de color. En algunos ejemplos, el elemento de sintaxis puede comprender un indicador de un bit. En algunos ejemplos, la unidad de codificación se codifica en un modo distinto del modo de modulación de código pulso intra (IPCM) y el elemento de sintaxis se señaliza sólo para unidades de codificación que utilizan un modo distinto del modo IPCM. En algunos ejemplos, el elemento de sintaxis indica que la unidad de codificación se codificó usando conversión de espacio de color cuando hay coeficientes

distintos de cero en una unidad de transformación de la unidad de codificación. En algunos ejemplos, un valor de 1 para el elemento de sintaxis indica que la unidad de codificación se codificó mediante conversión de espacio de color. La unidad 70 de decodificación de entropía del decodificador 30 de vídeo puede determinar además si un valor del elemento de sintaxis indica que la unidad de codificación se codificó usando conversión de espacio de color. En respuesta a la determinación de que el elemento de sintaxis indica que la unidad de codificación se codificó usando conversión de espacio de color, la unidad 70 de decodificación de entropía del decodificador 30 de vídeo puede aplicar el proceso de transformación inversa de espacio de color.

En algunos ejemplos, el elemento de sintaxis puede indicar que no se utilizó la conversión del espacio de color para codificar la unidad de codificación. Por ejemplo, el elemento de sintaxis puede indicar que no se usó conversión de espacio de color para codificar la unidad de codificación cuando la unidad de codificación está codificada intra y cuando un modo de predicción de luma y un modo de predicción de croma de una unidad de predicción dentro de la unidad de codificación son diferentes. En otro ejemplo, el elemento de sintaxis puede indicar que no se usó la conversión del espacio de color para codificar la unidad de codificación cuando la unidad de codificación está codificada con un modo de paleta. En estos ejemplos, el flujo de bits recibido puede no incluir el elemento de sintaxis.

En algunos ejemplos, el elemento de sintaxis es un primer elemento de sintaxis y los datos de vídeo son un primer conjunto de datos de vídeo. En tales ejemplos, la unidad 70 de decodificación de entropía del decodificador 30 de vídeo puede determinar además un valor de un segundo elemento de sintaxis de un segundo conjunto de datos de vídeo. El segundo elemento de sintaxis puede indicar si se usó un modo de paleta para codificar el segundo conjunto de datos de vídeo. En respuesta al segundo elemento de sintaxis que indica que se usó el modo de paleta para codificar los datos de vídeo, la unidad 70 de decodificación de entropía del decodificador 30 de vídeo puede determinar un valor de un tercer elemento de sintaxis. El tercer elemento de sintaxis puede indicar si se usa un primer modo o un segundo modo para decodificar un índice de paleta para un píxel en la unidad de codificación. En base al valor determinado del segundo elemento de sintaxis, la unidad 70 de decodificación de entropía del decodificador 30 de vídeo puede decodificar el índice de paleta usando el primer modo o el segundo modo. Cuando se utiliza el primer modo, la unidad 70 de decodificación de entropía del decodificador 30 de vídeo puede determinar un valor del índice de paleta, determinar un valor de un cuarto elemento de sintaxis que indica un número de píxeles en un orden de exploración inmediatamente posterior al píxel que se está decodificando actualmente, y duplique el resultado de determinar el valor del índice de paleta para los siguientes N píxeles en el orden de exploración, siendo N igual al valor del cuarto elemento de sintaxis. En algunos ejemplos, el primer modo es un modo de ejecución. Cuando se utiliza el segundo modo, la unidad 70 de decodificación de entropía del decodificador 30 de vídeo puede determinar el valor del índice de paleta y generar un valor de muestra de píxel para el píxel, donde el valor de muestra de píxel es igual al valor del índice de paleta. En algunos ejemplos, el segundo modo es un modo de píxel.

En algunos ejemplos, decodificar el elemento de sintaxis comprende que la unidad 70 de decodificación de entropía del decodificador 30 de vídeo esté configurada para decodificar datos de un mensaje de información de mejora suplementaria (SEI) que indica si la unidad de codificación se codificó usando conversión de espacio de color.

Se debe reconocer que dependiendo del ejemplo, ciertos actos o eventos de cualquiera de las técnicas descritas en el presente documento se pueden realizar en una secuencia diferente, se pueden agregar, fusionar u omitir por completo (por ejemplo, no todos los actos o eventos descritos son necesarios para la práctica de las técnicas). Además, en ciertos ejemplos, los actos o eventos pueden realizarse simultáneamente, por ejemplo, mediante procesamiento de múltiples subprocesos, procesamiento de interrupciones o múltiples procesadores, en lugar de secuencialmente.

En uno o más ejemplos, las funciones descritas pueden implementarse en hardware, software, firmware o cualquier combinación de los mismos. Si se implementan en software, las funciones pueden almacenarse o transmitirse como una o más instrucciones o código en un medio legible por ordenador y ejecutadas por una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que corresponden a un medio tangible tal como medios de almacenamiento de datos, o medios de comunicación que incluyen cualquier medio que facilite la transferencia de un programa informático de un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por ordenador generalmente pueden corresponder a (1) medios de almacenamiento tangibles legibles por ordenador que no son transitorios o (2) un medio de comunicación tal como una señal u onda portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible al que puedan acceder uno o más ordenadores o uno o más procesadores para recuperar instrucciones, códigos y/o estructuras de datos para la implementación de las técnicas descritas en esta divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

A modo de ejemplo, y sin limitación, dichos medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento en disco óptico, almacenamiento en disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que se puede utilizar para almacenar el código de programa deseado en forma de instrucciones o estructuras de datos y al que se puede acceder mediante un ordenador. Además, cualquier conexión se denomina propiamente un medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde un sitio web, servidor u otra fuente remota utilizando un cable coaxial, cable de fibra óptica, par trenzado, línea de abonado digital (DSL) o tecnologías inalámbricas tales como infrarrojos, radio y

microondas, entonces el cable coaxial, el cable de fibra óptica, el par trenzado, el DSL o las tecnologías inalámbricas tales como infrarrojos, radio y microondas se incluyen en la definición de medio. Sin embargo, debe entenderse que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino que están dirigidos a medios de almacenamiento tangibles y no transitorios. Disco y disco, tal como se utilizan en el presente documento, incluyen discos compactos (CD), discos láser, discos ópticos, discos versátiles digitales (DVD), disquetes y discos Blu-ray, donde los discos generalmente reproducen datos magnéticamente, mientras que los discos reproducen datos ópticamente con láseres. Las combinaciones de lo anterior también deben incluirse dentro del alcance de los medios legibles por ordenador.

Las instrucciones pueden ser ejecutadas por uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de uso general, circuitos integrados de aplicaciones específicas (ASIC), matrices lógicas programables en campo (FPGA) u otros circuitos lógicos integrados o discretos equivalentes. Por consiguiente, el término "procesador", tal como se utiliza en el presente documento, puede referirse a cualquiera de las estructuras anteriores o cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento puede proporcionarse dentro de módulos de hardware y/o software dedicados configurados para codificar y decodificar, o incorporados en un códec combinado. Además, las técnicas podrían implementarse completamente en uno o más circuitos o elementos lógicos.

Las técnicas de esta divulgación pueden implementarse en una amplia variedad de dispositivos o aparatos, incluyendo un teléfono inalámbrico, un circuito integrado (CI) o un conjunto de CI (por ejemplo, un conjunto de chips). En esta divulgación se describen diversos componentes, módulos o unidades para enfatizar aspectos funcionales de los dispositivos configurados para realizar las técnicas divulgadas, pero no necesariamente requieren la realización por diferentes unidades de hardware. Más bien, como se describió anteriormente, se pueden combinar varias unidades en una unidad de hardware de códec o proporcionarse mediante una colección de unidades de hardware interoperativas, incluyendo uno o más procesadores como se describió anteriormente, junto con software y/o firmware adecuado.

Se han descrito diversos ejemplos de la divulgación. El alcance de la protección está definido por las siguientes reivindicaciones.

REIVINDICACIONES

1. Un método para codificar datos de vídeo intrapredichos, comprendiendo los datos de vídeo intra predichos datos residuales que representan diferencias de píxeles entre un bloque original que se va a codificar y un bloque predictivo, comprendiendo el método:

para el modo de codificación con y sin pérdidas, aplicar (158) un proceso directo de transformación del espacio de color YCoCg-R a los datos residuales; y
cuando se aplica codificación con pérdida, modificar los datos residuales transformados escalando los componentes croma mediante un desplazamiento a la derecha de N bits, de modo que la precisión de profundidad de bits de los componentes luma y croma sea igual.

2. El método de la reivindicación 1, que comprende además determinar el uso de un proceso de transformación del espacio de color para codificar los datos de vídeo, en el que determinar el uso de un proceso de transformación del espacio de color para codificar los datos de vídeo comprende:

establecer un valor de un elemento de sintaxis que indica que los datos de vídeo se codificaron mediante un proceso de transformación del espacio de color,
en el que el elemento de sintaxis comprende preferiblemente un indicador de un bit, y en el que un valor de 1 para el elemento de sintaxis indica preferiblemente que se utiliza un proceso de transformación del espacio de color para codificar los datos de vídeo.

3. El método de la reivindicación 2, en el que para una unidad de codificación codificada intra, el elemento de sintaxis se señala sólo cuando la unidad de codificación se codifica utilizando un modo distinto del modo de modulación de código pulso intra (IPCM); y/o

en el que el elemento de sintaxis se señala sólo cuando hay coeficientes distintos de cero en una unidad de transformación de la unidad de codificación; y/o

en el que el elemento de sintaxis no se señala cuando la unidad de codificación está codificada intra y cuando un modo de predicción de luma y un modo de predicción de croma de una unidad de predicción dentro de la unidad de codificación son diferentes; y/o

en el que el elemento de sintaxis no se señala cuando la unidad de codificación se codifica con un modo de paleta.

4. Un dispositivo de codificación de vídeo que comprende:

una memoria configurada para almacenar datos de vídeo intrapredichos, comprendiendo los datos de vídeo intrapredichos datos residuales que representan diferencias de píxeles entre un bloque original que se va a codificar y un bloque predictivo; y

uno o más procesadores configurados para:

para el modo de codificación con y sin pérdidas, aplique un proceso de transformación directa del espacio de color YCoCg-R a los datos residuales; y

cuando se aplica codificación con pérdida, modificar los datos residuales transformados mediante escalar los componentes de croma mediante un desplazamiento a la derecha de N bits, de modo que las precisiones de profundidad de bits de los componentes de luma y croma sean iguales.

5. El dispositivo de codificación de vídeo de la reivindicación 4, en el que uno o más procesadores están configurados además para determinar el uso de un proceso de transformación del espacio de color para codificar los datos de vídeo, en el que uno o más procesadores están configurados para determinar el uso de un proceso de transformación del espacio de color para codificar los datos de vídeo, comprende uno o más procesadores configurados para:

establecer un valor de un elemento de sintaxis que indica que los datos de vídeo se codificaron usando un proceso de transformación de espacio de color, en donde el elemento de sintaxis comprende preferiblemente una bandera de un bit, y en donde un valor de 1 para el elemento de sintaxis indica preferiblemente que se utiliza un proceso de transformación del espacio de color para codificar los datos de vídeo.

6. El dispositivo de codificación de vídeo de la reivindicación 5, en el que el elemento de sintaxis se señala, para una unidad de codificación codificada intra, sólo cuando la unidad de codificación está codificada utilizando un modo distinto del modo de modulación de código pulso intra (IPCM); y/o

en el que el elemento de sintaxis se señala sólo cuando hay coeficientes distintos de cero en una unidad de transformación de la unidad de codificación; y/o

en el que el elemento de sintaxis no se señala cuando la unidad de codificación está codificada intra y cuando un modo de predicción de luma y un modo de predicción de croma de una unidad de predicción dentro de la unidad de codificación son diferentes; y/o

en el que el elemento de sintaxis no se señala cuando la unidad de codificación se codifica con un modo de paleta.

7. Un medio de almacenamiento legible por ordenador que comprende instrucciones que, cuando se ejecutan, hacen que uno o más procesadores de un dispositivo de codificación o decodificación de vídeo realicen el método de cualquiera de las reivindicaciones 1-3 u 8-11.

5 8. Un método para decodificar datos de vídeo codificados intra, comprendiendo los datos de vídeo datos residuales que representan diferencias de píxeles entre un bloque original que se va a codificar y un bloque predictivo, comprendiendo el método:

10 cuando se aplica codificación con pérdida, modificar (212) los datos residuales escalando los componentes de croma mediante un desplazamiento a la izquierda de N bits, de modo que la precisión de profundidad de bits de los componentes de luma y croma sea igual; y
aplicar (214) un proceso de transformación de espacio de color YCoCg-R inverso al residuo modificado
datos, en el que el proceso de transformación inversa del espacio de color YCoCg-R se aplica tanto para los modos de codificación con pérdida como sin pérdida.

15 9. El método de la reivindicación 8, que comprende además:

20 decodificar un elemento de sintaxis de una unidad de codificación de datos de vídeo, en donde el elemento de sintaxis indica si la unidad de codificación se codificó usando un proceso de transformación de espacio de color;
determinar si un valor del elemento de sintaxis indica que la unidad de codificación se codificó usando un proceso de transformación de espacio de color; y
en respuesta a determinar que el elemento de sintaxis indica que la unidad de codificación se codificó usando un proceso de transformación del espacio de color, aplicando el proceso de transformación del espacio de color inverso,
25 en el que el elemento de sintaxis comprende un indicador de un bit, y en el que un valor de 1 para el elemento de sintaxis indica preferiblemente que se utiliza un proceso de transformación del espacio de color para codificar los datos de vídeo.

30 10. El método de la reivindicación 9, en el que el elemento de sintaxis se señala sólo para unidades de codificación codificadas intra que utilizan un modo distinto del modo IPCM; y/o

en el que el elemento de sintaxis se señala sólo cuando hay coeficientes distintos de cero en una unidad de transformación de la unidad de codificación; y/o
en el que el elemento de sintaxis indica que la unidad de codificación no se codificó utilizando un proceso de transformación del espacio de color cuando la unidad de codificación está codificada intra y cuando un modo de predicción de luma y un modo de predicción de croma de una unidad de predicción dentro de la unidad de codificación son diferentes, en el que el elemento de sintaxis no está presente en un flujo de bits recibido que comprende los datos de vídeo, y en el que decodificar el elemento de sintaxis comprende inferir el valor del elemento de sintaxis; y/o
35 en donde el elemento de sintaxis indica que la unidad de codificación no se codificó usando un proceso de transformación de espacio de color cuando la unidad de codificación está codificada con un modo de paleta, en donde
40 el elemento de sintaxis no está presente en un flujo de bits recibido que comprende los datos de vídeo, y en donde decodificar el elemento de sintaxis comprende inferir el valor del elemento de sintaxis.

45 11. El método de la reivindicación 1 o la reivindicación 8, en el que los datos de vídeo son un primer conjunto de datos de vídeo, comprendiendo además el método:

para un segundo conjunto de datos de vídeo:
determinar una profundidad de bits de un componente luma del segundo conjunto de datos de vídeo y una profundidad de bits de un componente croma del segundo conjunto de datos de vídeo; y
50 en respuesta a que la profundidad de bits del componente luma del segundo conjunto de datos de vídeo sea diferente de la profundidad de bits del componente croma del segundo conjunto de datos de vídeo, deshabilitar la transformación del espacio de color o la transformación del espacio de color inversa para el segundo conjunto de datos de vídeo,

el método comprende además preferiblemente:

55 cuando la reivindicación depende de la reivindicación 8, decodificar un segundo elemento de sintaxis de un conjunto de parámetros de imagen del segundo conjunto de datos de vídeo que indica que el proceso de transformación inversa del espacio de color está deshabilitado para el segundo conjunto de datos de vídeo, o
cuando la reivindicación depende de la reivindicación 1,
establecer un segundo elemento de sintaxis de un conjunto de parámetros de imagen del segundo conjunto de datos de vídeo que indica que el proceso de transformación del espacio de color está desactivado para el segundo conjunto de datos de vídeo.
60

12. Un dispositivo de decodificación de vídeo que comprende:

una memoria configurada para almacenar datos de vídeo codificados intra, comprendiendo los datos de vídeo datos residuales que representan diferencias de píxeles entre un bloque original que se va a codificar y un bloque predictivo; y

uno o más procesadores configurados para:

5 cuando se aplica codificación con pérdida, modificar los datos residuales escalando los componentes croma mediante un desplazamiento a la izquierda de N bits, de modo que la precisión de profundidad de bits de los componentes luma y croma sea igual; y

10 aplicar un proceso de transformación inversa del espacio de color YCoCg-R a los datos residuales modificados, en donde el procesador está configurado para aplicar el proceso de transformación inversa del espacio de color YCoCg-R para modos de codificación con y sin pérdidas.

13. El dispositivo de decodificación de vídeo de la reivindicación 12, en el que uno o más procesadores están configurados además para:

15 decodificar un elemento de sintaxis de una unidad de codificación de datos de vídeo, en donde el elemento de sintaxis indica si la unidad de codificación se codificó usando un proceso de transformación de espacio de color; determinar si un valor del elemento de sintaxis indica que la unidad de codificación se codificó usando un proceso de transformación de espacio de color; y

20 en respuesta a la determinación de que el elemento de sintaxis indica que la unidad de codificación se codificó usando un proceso de transformación del espacio de color, aplicar el proceso de transformación del espacio de color inverso, en el que el elemento de sintaxis comprende preferiblemente un indicador de un bit, y en el que un valor de 1 para el elemento de sintaxis indica preferiblemente que se utiliza un proceso de transformación del espacio de color para codificar los datos de vídeo.

25 14. El dispositivo de decodificación de vídeo de la reivindicación 13, en el que el elemento de sintaxis se señala sólo para unidades de codificación codificadas intra que utilizan un modo distinto del modo IPCM; y/o

en el que el elemento de sintaxis se señala sólo cuando hay coeficientes distintos de cero en una unidad de transformación de la unidad de codificación; y/o

30 en el que el elemento de sintaxis indica que la unidad de codificación no se codificó utilizando un proceso de transformación del espacio de color cuando la unidad de codificación está codificada intra y cuando un modo de predicción de luma y un modo de predicción de croma de una unidad de predicción dentro de la unidad de codificación son diferentes, en el que el elemento de sintaxis no está presente en un flujo de bits recibido que comprende los datos de vídeo, y en el que decodificar el elemento de sintaxis comprende inferir el valor del elemento de sintaxis; y/o

35 en donde el elemento de sintaxis indica que la unidad de codificación no se codificó usando un proceso de transformación de espacio de color cuando la unidad de codificación está codificada con un modo de paleta, en donde el elemento de sintaxis no está presente en un flujo de bits recibido que comprende los datos de vídeo, y en donde decodificar el elemento de sintaxis comprende inferir el valor del elemento de sintaxis.

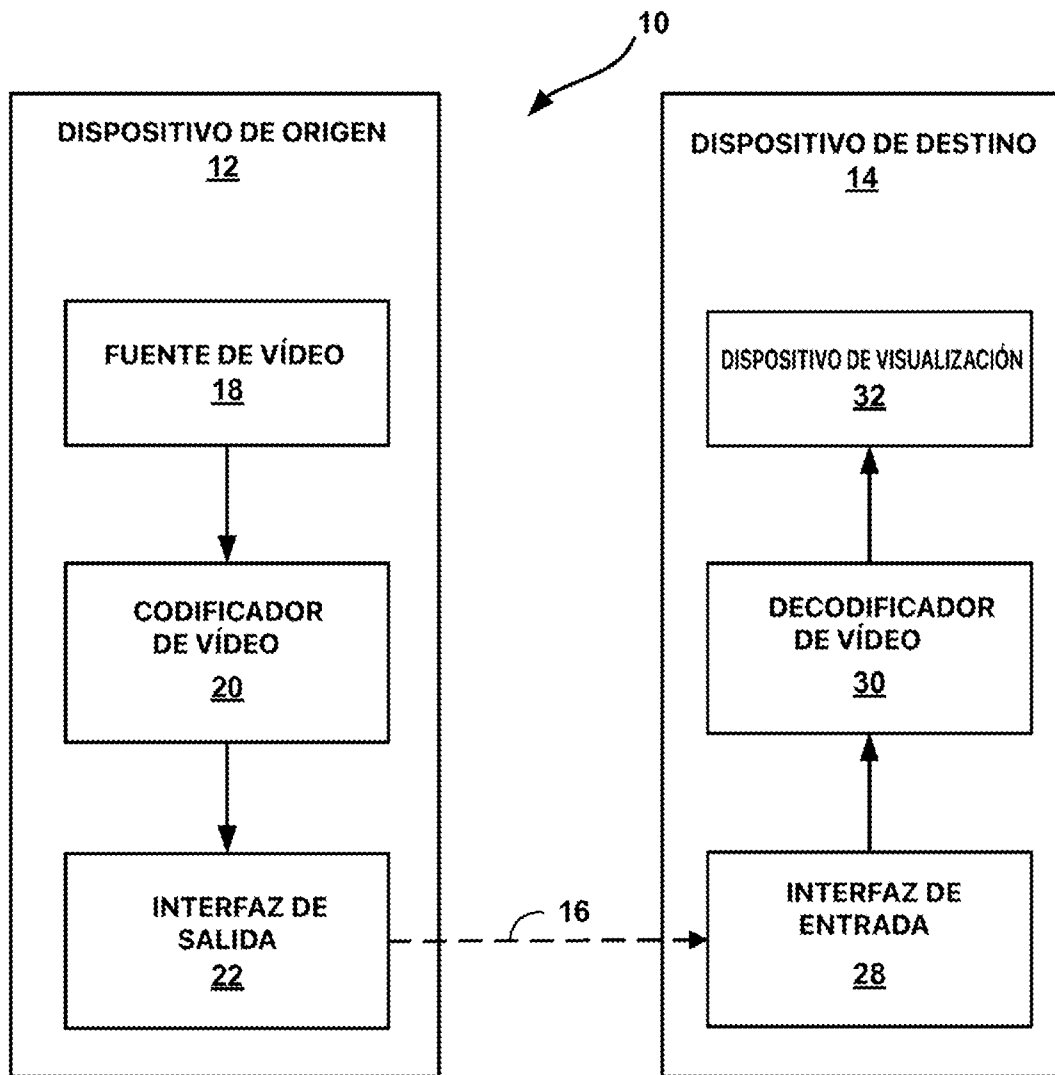


FIG. 1

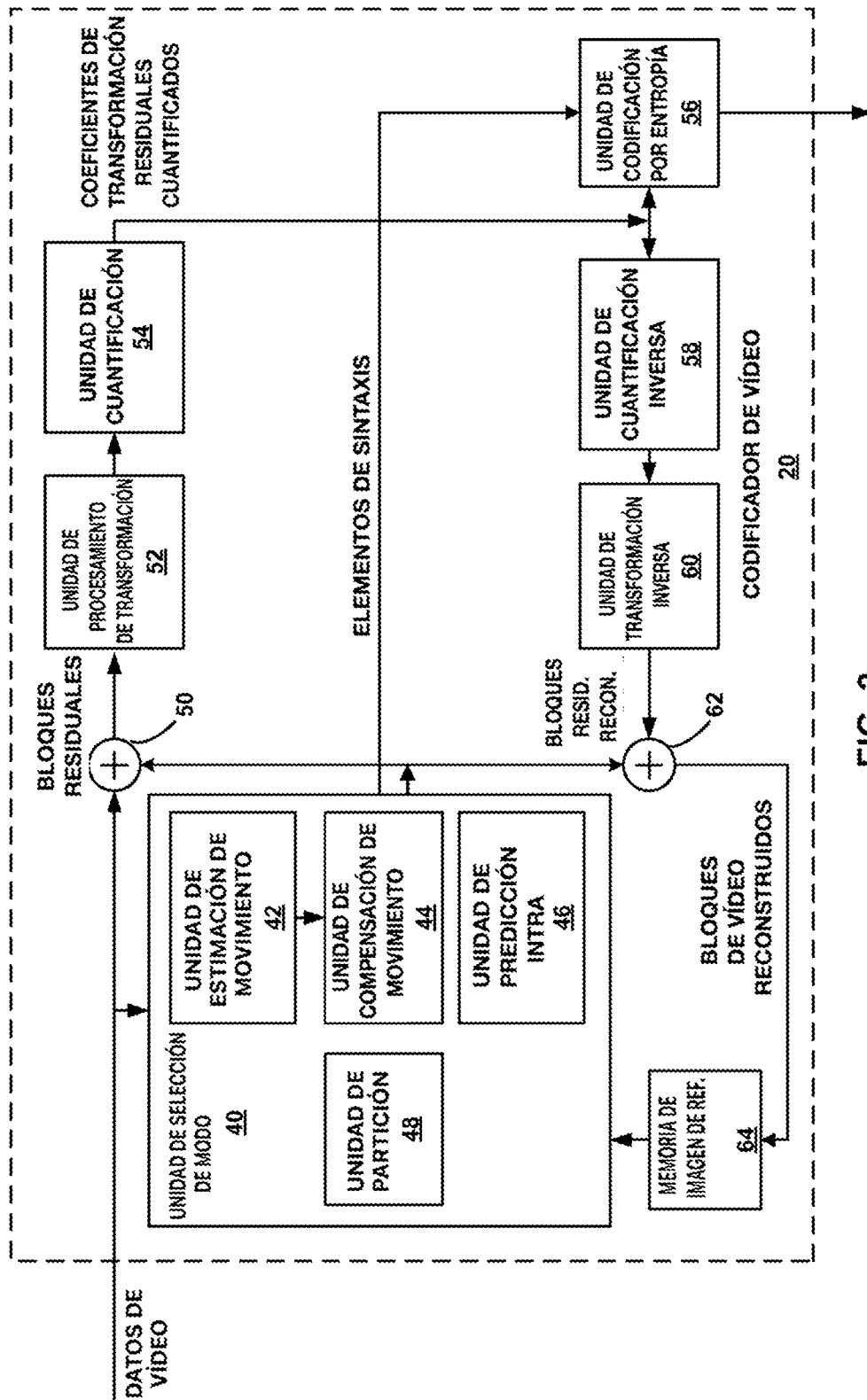


FIG. 2

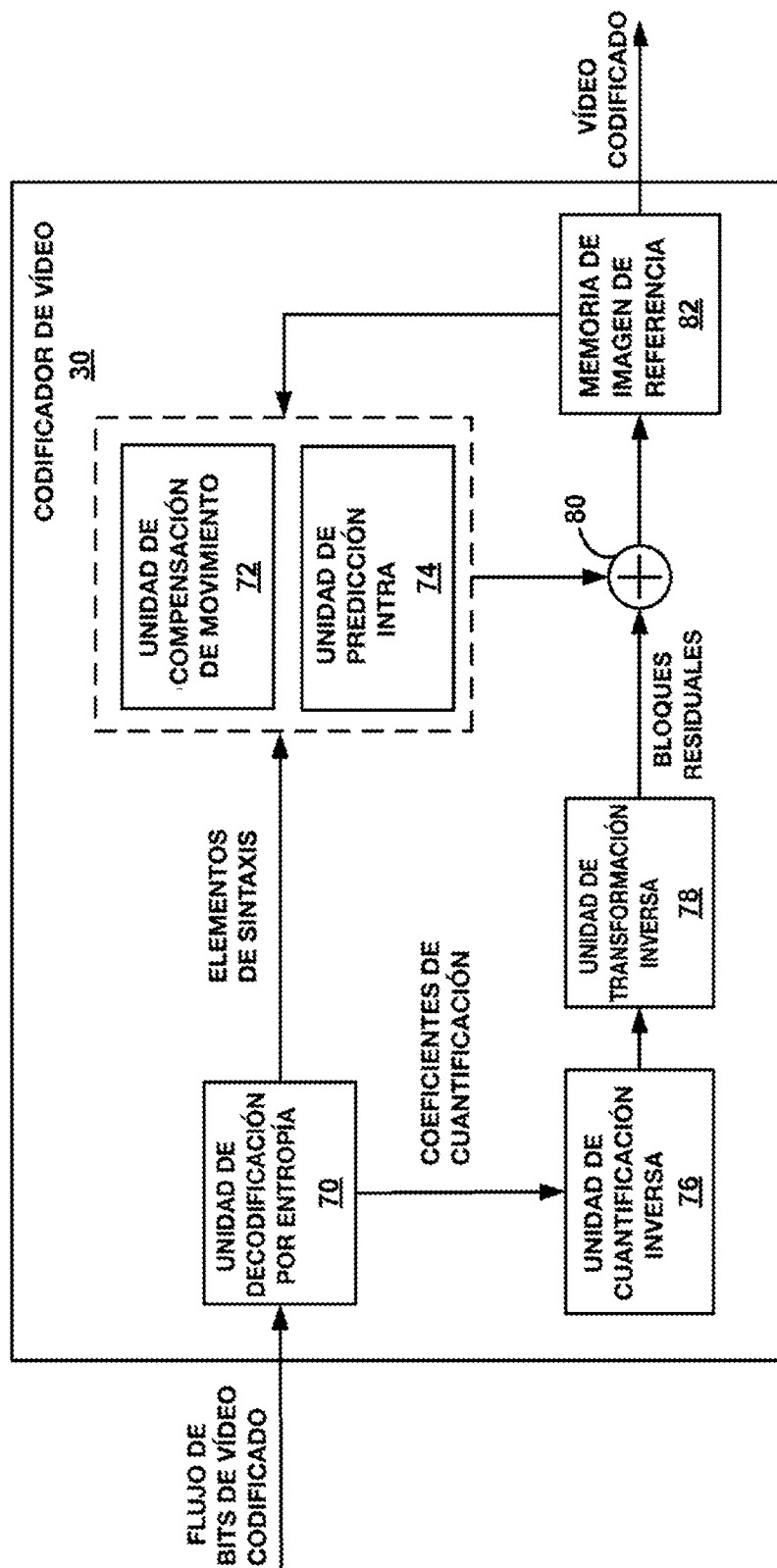


FIG. 3

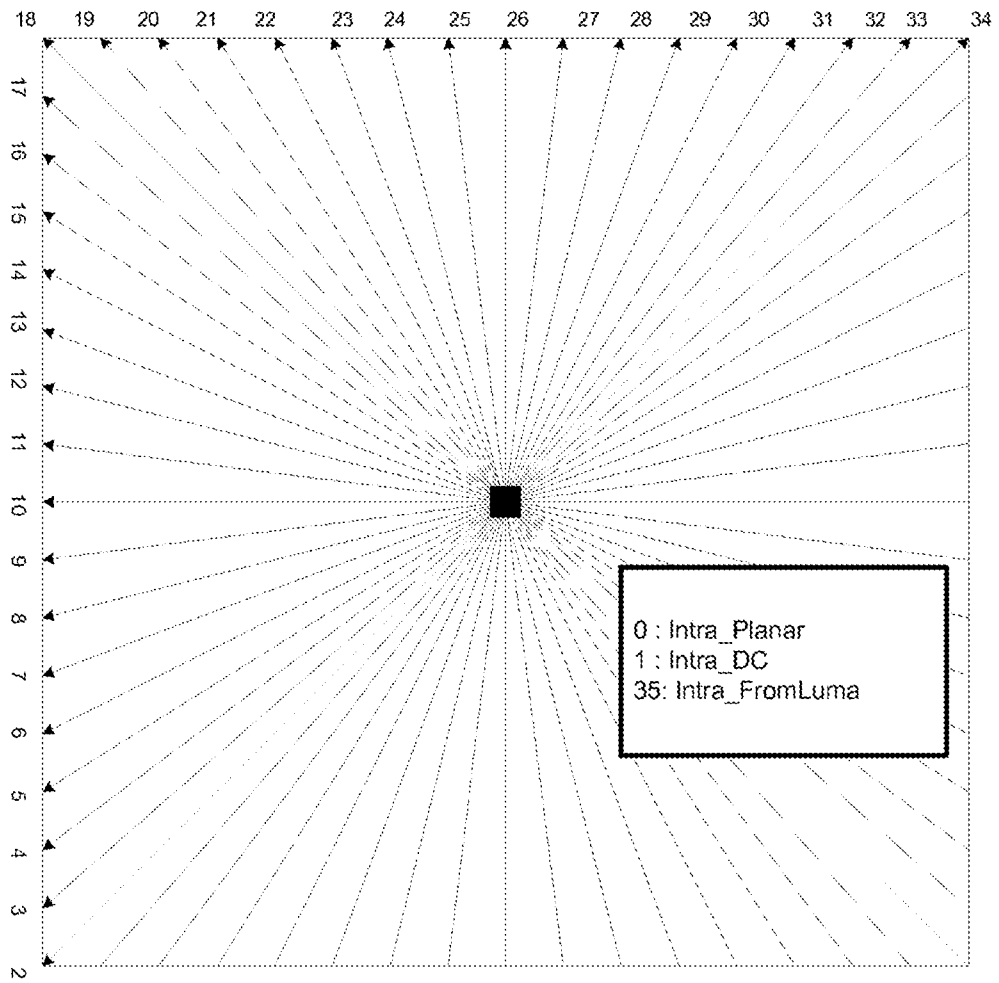


FIG. 4

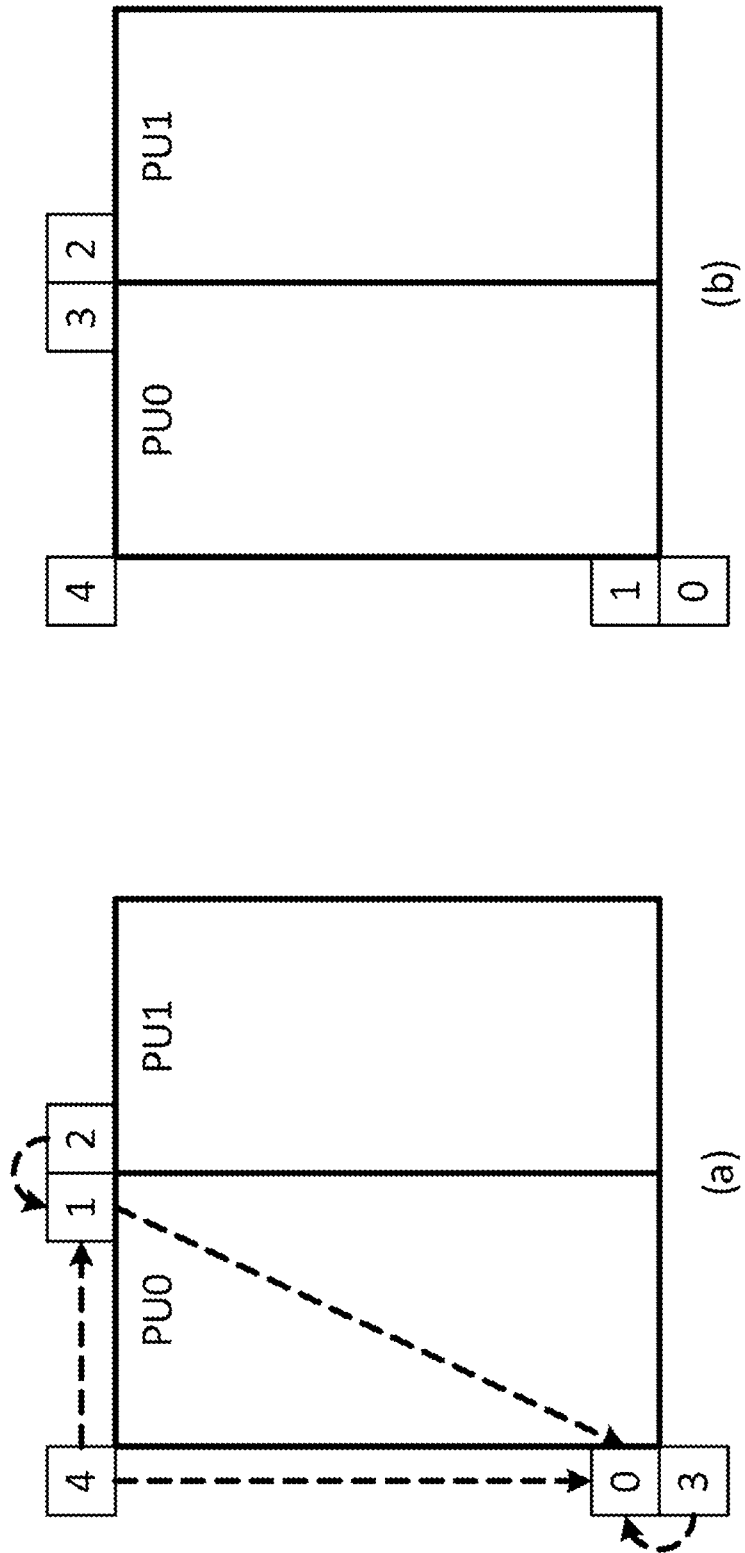


FIG. 5

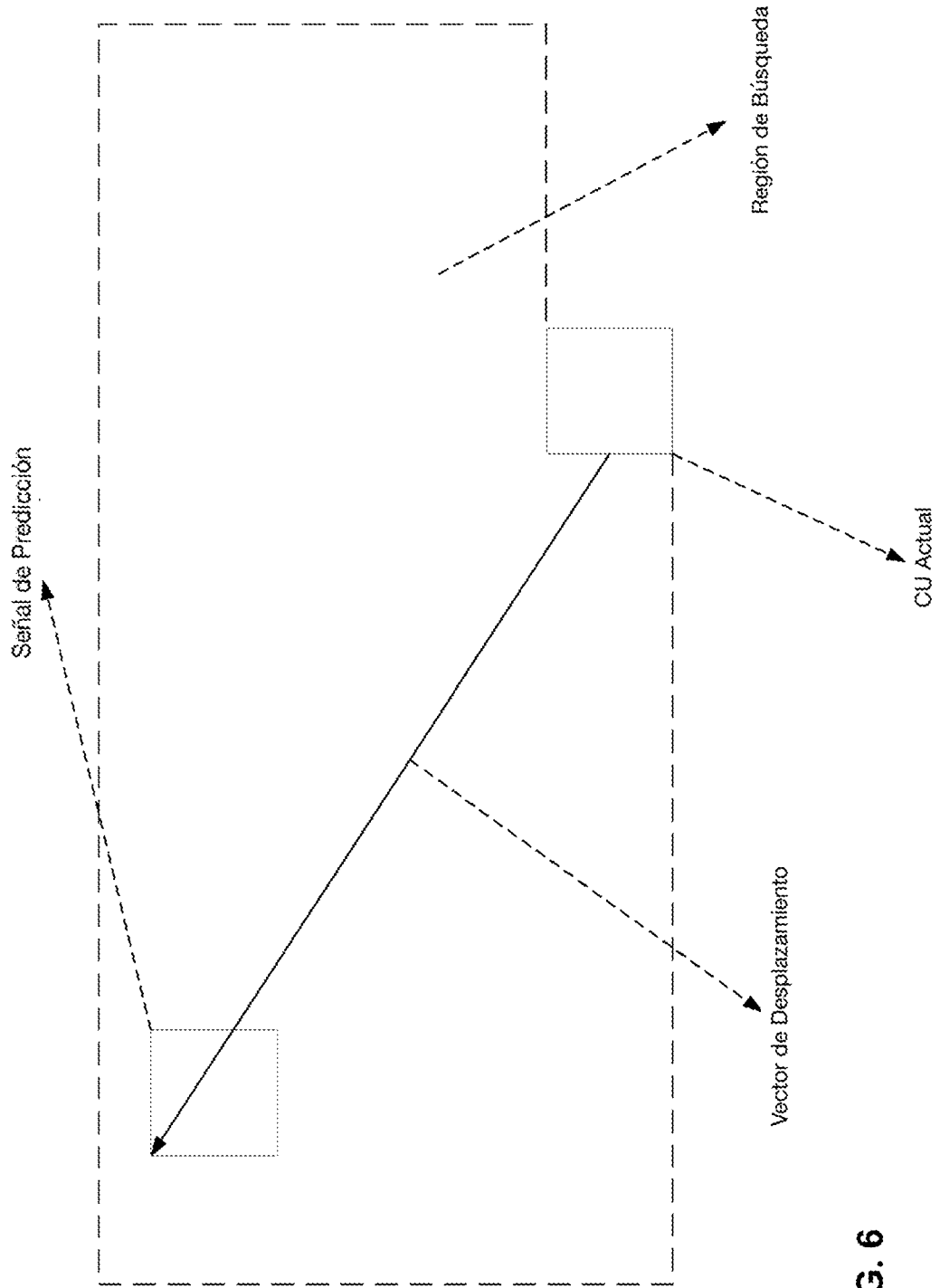


FIG. 6

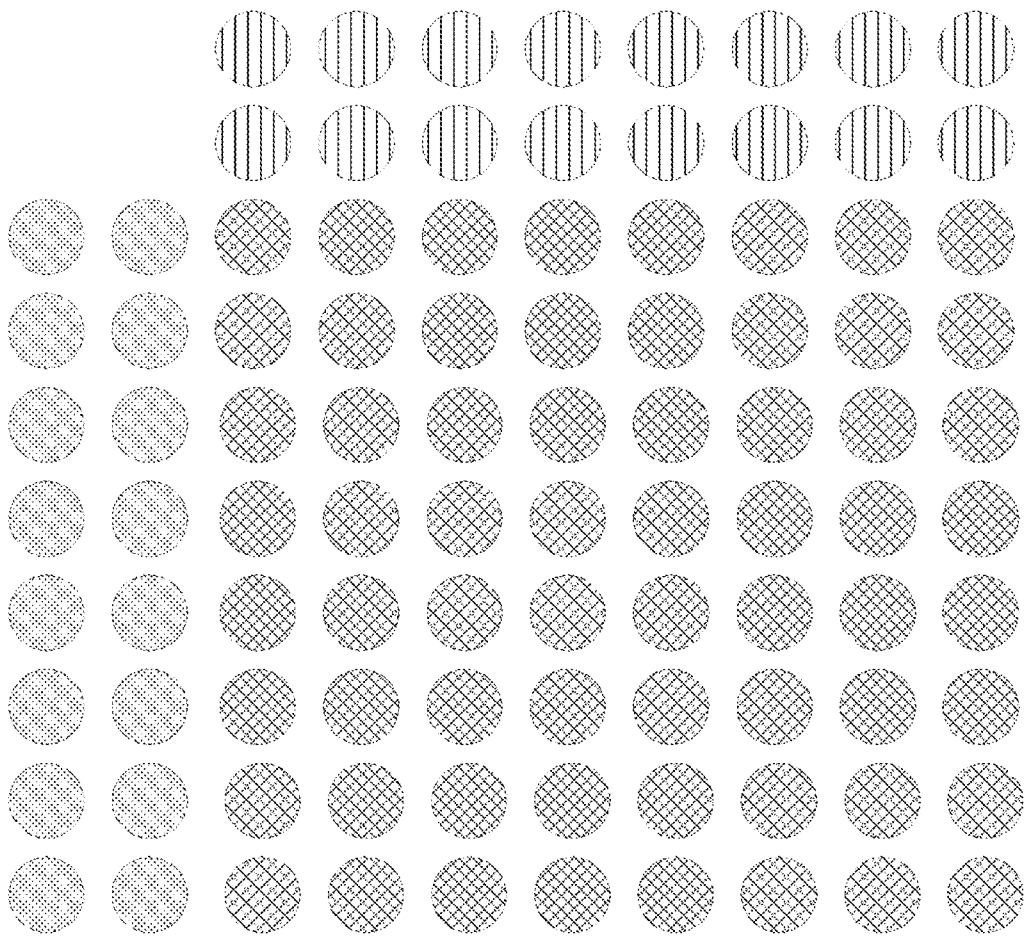


FIG. 7

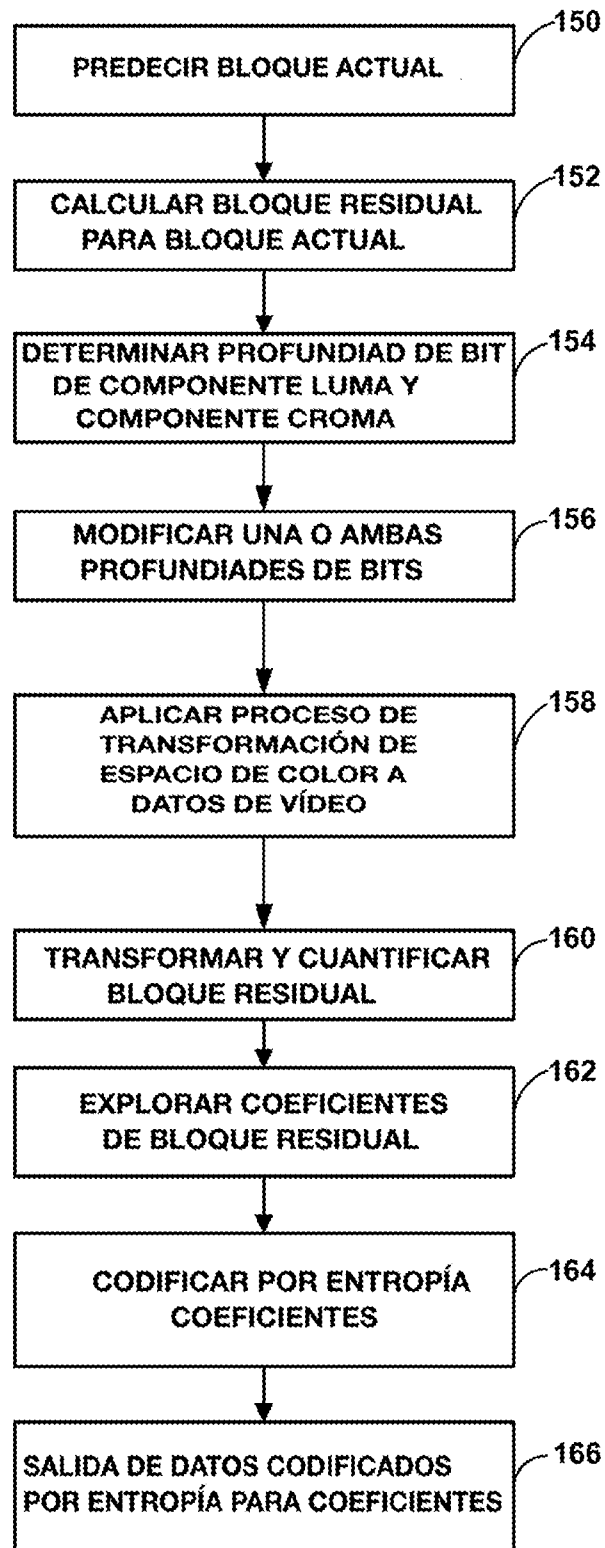


FIG. 8

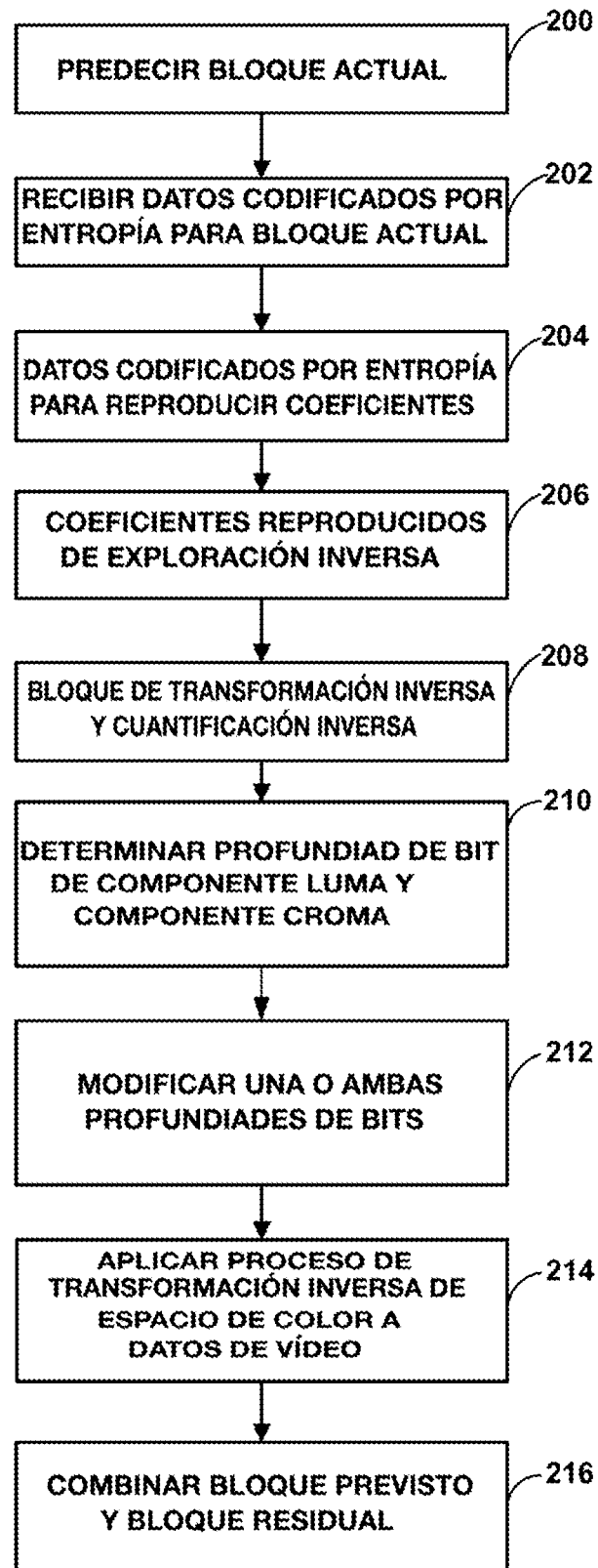


FIG. 9