

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 778 463**

51 Int. Cl.:

H04N 19/103 (2014.01)
H04N 19/70 (2014.01)
H04N 19/172 (2014.01)
H04N 19/423 (2014.01)
H04N 19/186 (2014.01)
H04N 19/187 (2014.01)
H04N 19/146 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 86 Fecha de presentación y número de la solicitud internacional: **30.12.2014 PCT/US2014/072705**
 87 Fecha y número de publicación internacional: **09.07.2015 WO15103231**
 96 Fecha de presentación y número de la solicitud europea: **30.12.2014 E 14825075 (6)**
 97 Fecha y número de publicación de la concesión europea: **18.12.2019 EP 3090557**

54 Título: **Inferencia mejorada de indicador de no emisión de imagen previa en codificación de video**

30 Prioridad:

03.01.2014 US 201461923530 P
29.12.2014 US 201414584351

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
10.08.2020

73 Titular/es:

QUALCOMM INCORPORATED (100.0%)
5775 Morehouse Drive
San Diego, CA 92121-1714, US

72 Inventor/es:

WANG, YE-KUI y
CHEN, YING

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 778 463 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

Inferencia mejorada de indicador de no emisión de imagen previa en codificación de video

5 CAMPO TÉCNICO

10 **[0001]** Esta divulgación se refiere al campo de la codificación y compresión de video, que incluye tanto la codificación de video de una sola capa como la codificación de video de múltiples capas. La codificación de video de múltiples capas puede incluir codificación de video escalable (SVC), codificación de video de vista múltiple (MVC), codificación de video 3D (3DV), etc.

ANTECEDENTES

15 **[0002]** Las capacidades del vídeo digital se pueden incorporar a una amplia gama de dispositivos, incluyendo televisores digitales, sistemas de radiodifusión directa digital, sistemas de radiodifusión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de sobremesa, cámaras digitales, dispositivos de grabación digitales, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos celulares o de radio por satélite, dispositivos de videoconferencia y similares. Los dispositivos de vídeo digital implementan técnicas de compresión de vídeo, tales como las descritas en las normas definidas por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Parte 10, Codificación Avanzada de Vídeo (AVC), la norma de Codificación de Vídeo de Alta Eficacia (HEVC) y las ampliaciones de dichas normas. Los dispositivos de vídeo pueden transmitir, recibir, codificar, descodificar y/o almacenar información de vídeo digital más eficazmente, implementando dichas técnicas de codificación de vídeo.

20 **[0003]** Las técnicas de compresión de vídeo realizan predicción espacial (intraimagen) y/o predicción temporal (interimagen) para reducir o eliminar la redundancia intrínseca a las secuencias de vídeo. Para la codificación de vídeo basada en bloques, un fragmento de vídeo (por ejemplo, una trama de vídeo, una parte de una trama de vídeo, etc.) 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 un fragmento intracodificado (I) de una imagen se codifican usando predicción espacial con respecto a muestras de referencia en bloques vecinos en la misma imagen. Los bloques de vídeo de un fragmento intercodificado (P o B) de una imagen pueden usar predicción espacial con respecto a unas muestras de referencia de bloques vecinos de la misma imagen, o predicción temporal con respecto a unas muestras de referencia de otras imágenes de referencia. Las imágenes pueden denominarse tramas, y las imágenes de referencia pueden denominarse tramas de referencia.

25 **[0004]** 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 a codificar y el bloque predictivo. Un bloque intercodificado 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 que indican la diferencia entre el bloque codificado y el bloque predictivo. Un bloque intracodificado se codifica de acuerdo con un modo de intracodificación y de acuerdo con datos residuales. Para una mayor compresión, los datos residuales pueden transformarse desde el dominio del píxel a un dominio de transformada, dando como resultado coeficientes de transformada residuales, que a continuación se pueden cuantificar. Los coeficientes de transformación cuantificados, dispuestos inicialmente en una matriz bidimensional, pueden escanearse con el fin de generar un vector unidimensional de coeficientes de transformación, y puede aplicarse codificación por entropía para lograr aún más compresión.

30 **[0005]** RAMASUBRAMONIAN AK ET AL: "MV-HEVC/SHVC HLS: On flushing of decoded pictures from the DPB based on NoOutputOfPriorPicsFlag", 15. REUNIÓN JCT-VC; 23-10-2013 - 1-11-2013; GINEBRA; (EQUIPO COLABORATIVO CONJUNTO SOBRE LA CODIFICACIÓN DE VIDEO DE ISO/CEI JTC1/SC29/WG11 Y UIT-T SG.16); URL: HTTP://WFTP3,ITU.INT/AV-ARCH/JCTVC-SITE/, no. JCTVC-O0266, 15 de octubre de 2013 (2013-10-15), XP030115331 , revela el vaciado específico de capas de imágenes decodificadas de la memoria intermedia de imágenes decodificadas (DPB) para imágenes de puntos de acceso intra aleatorios (IRAP). El indicador NoOutputOfPriorPicsFlag indica si las imágenes previamente decodificadas en un DPB deben eliminarse sin salir. Este indicador se establece en función de la resolución de luminancia espacial y del número de almacenes de imágenes de la imagen actual y la imagen previamente decodificada de la misma capa.

SUMARIO

35 **[0006]** La codificación de vídeo escalable (SVC) se refiere a la codificación de vídeo en la que se utilizan una capa base (BL), a veces denominada capa de referencia (RL), y una o más capas de mejora escalable (EL). Para SVC, la capa base puede transportar datos de vídeo con un nivel de calidad base. Las una o más capas de mejora pueden transportar datos de vídeo adicionales para admitir, por ejemplo, niveles espaciales, temporales y/o de señal a ruido (SNR) más altos. Las capas de mejora pueden definirse en relación con una capa previamente codificada. Por ejemplo, una capa inferior puede servir como una BL, mientras que una capa superior puede servir como una EL. Las capas intermedias pueden servir como EL o RL, o como ambas. Por ejemplo, una capa intermedia (por ejemplo, una capa que ni es la capa más baja ni la capa más alta) puede ser una EL para las capas debajo de la capa intermedia, como

la capa base o cualquier capa de mejora intermedia, y al mismo tiempo servir como RL para una o más capas de mejora sobre la capa intermedia. De manera similar, en la ampliación de múltiples vistas o 3D de la norma HEVC, puede haber múltiples vistas, y la información de una vista puede utilizarse para codificar (por ejemplo, codificar o descodificar) la información de otra vista (por ejemplo, estimación de movimiento, predicción del vector de movimiento y/u otras redundancias).

[0007] Un aparato para codificar información de vídeo de acuerdo con ciertos aspectos incluye una memoria y un procesador. La unidad de memoria está configurada para almacenar información de vídeo asociada con una capa actual. El procesador está configurado para: obtener al menos uno de los formatos de croma de una imagen actual de la capa actual que se va a decodificar, una profundidad de bits de muestras de luma de la imagen actual o una profundidad de bits de muestras de croma de la imagen actual, en el que la imagen actual es una imagen de punto de acceso intra aleatorio (IRAP) que inicia una nueva secuencia de vídeo codificada (CVS); obtener al menos uno de los formatos de croma de una imagen anterior de la capa actual que se decodificó, una profundidad de bits de muestras de luma de la imagen anterior, o una profundidad de bits de muestras de croma de la imagen anterior; y determinar un valor de un primer indicador asociado con la imagen actual, el primer indicador indica si las imágenes en una memoria intermedia de imagen decodificada (DPB) deben emitirse, en el que la determinación del valor del primer indicador se basa en al menos uno de: (1) el formato de croma de la imagen actual y el formato de croma de la imagen anterior, (2) la profundidad de bits de las muestras de luma de la imagen actual y la profundidad de bits de las muestras de luma de la imagen anterior, o (3) la profundidad de bits de las muestras de croma de la imagen actual y la profundidad de bits de las muestras de croma de la imagen anterior.

[0008] Cada uno de los sistemas, procedimientos y dispositivos de esta divulgación tiene varios aspectos innovadores, ninguno de los cuales es el único responsable de los atributos deseables divulgados en el presente documento. Los detalles de uno o más ejemplos se exponen en los dibujos adjuntos y en la descripción. Otras características, objetos y ventajas resultarán evidentes a partir de la descripción y de los dibujos, y a partir de las reivindicaciones.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

[0009]

La FIG. 1A es un diagrama de bloques que ilustra un sistema de codificación y descodificación de vídeo de ejemplo que puede usar técnicas según los aspectos descritos en esta divulgación.

La FIG. 1B es un diagrama de bloques que ilustra otro sistema de codificación y descodificación de vídeo de ejemplo que puede llevar a cabo técnicas según los aspectos descritos en esta divulgación.

La FIG. 2A es un diagrama de bloques que ilustra un ejemplo de un codificador de vídeo que puede implementar técnicas de acuerdo con aspectos descritos en esta divulgación.

La FIG. 2B es un diagrama de bloques que ilustra un ejemplo de un codificador de vídeo que puede implementar técnicas de acuerdo con aspectos descritos en esta divulgación.

La FIG. 3A es un diagrama de bloques que ilustra un ejemplo de un descodificador de vídeo que puede implementar técnicas de acuerdo con aspectos descritos en esta divulgación.

La FIG. 3B es un diagrama de bloques que ilustra un ejemplo de un descodificador de vídeo que puede implementar técnicas de acuerdo con aspectos descritos en esta divulgación.

La FIG. 4 es un diagrama de bloques que ilustra una configuración de ejemplo de imágenes en diferentes capas de acuerdo con un modo de realización de la presente divulgación.

La FIG. 5 es un diagrama de flujo que ilustra un procedimiento de codificación de información de vídeo, de acuerdo con un modo de realización de la presente divulgación.

DESCRIPCIÓN DETALLADA

[0010] En general, esta divulgación se refiere a codificación de una sola capa, así como a la predicción entre capas para la codificación de vídeo escalable en el contexto de los códecs de vídeo avanzados, como HEVC (codificación de vídeo de alta eficiencia). Más específicamente, la presente divulgación se refiere a sistemas y procedimientos para mejorar el rendimiento de la predicción entre capas en extensión de codificación de vídeo escalable de HEVC, que se puede denominar como SHVC.

[0011] En la descripción a continuación, se describen técnicas de codificación avanzada de vídeo (AVC)/H.264 relacionadas con ciertos modos de realización; también se analizan la norma HEVC y las técnicas relacionadas. Aunque ciertos modos de realización se describen en el presente documento en el contexto de las normas HEVC y/o

H.264, un experto en la materia puede apreciar que los sistemas y procedimientos divulgados en el presente documento pueden aplicarse a cualquier norma de codificación de vídeo adecuada. Por ejemplo, los modos de realización divulgados en el presente documento pueden ser aplicables a una o más de las siguientes normas: Sector de Normalización de las Telecomunicaciones de la Unión Internacional de Telecomunicaciones (UIT) (UIT-T) H.261, Organización Internacional de Normalización (ISO) y Grupo de Expertos en Imágenes en Movimiento (MPEG) 1 (MPEG-1) de la Comisión Electrotécnica Internacional (IEC) (ISO/IEC) Visual, ITU-T H.262 o ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO / IEC MPEG-4 Visual y 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).

[0012] HEVC, en general, sigue el marco de las normas de codificación de vídeo anteriores en muchos aspectos. La unidad de predicción en HEVC es diferente de las unidades de predicción (por ejemplo, macrobloque) en ciertas normas anteriores de codificación de vídeo. De hecho, el concepto de un macrobloque no existe en HEVC como se entiende en ciertas normas anteriores de codificación de vídeo. Un macrobloque se reemplaza por una estructura jerárquica basada en un esquema de árbol cuaternario, que puede proporcionar una gran flexibilidad, entre otros beneficios posibles. Por ejemplo, dentro del esquema HEVC, se definen tres tipos de bloques, unidad de codificación (CU), unidad de predicción (PU) y unidad de transformación (TU). La CU puede referirse a la unidad básica de división de la región. La CU puede considerarse análoga al concepto de macrobloque, pero HEVC no limita el tamaño máximo de las CU y puede permitir la división recursiva en cuatro CU de igual tamaño para mejorar la adaptabilidad del contenido. La PU puede considerarse la unidad básica de inter/intrapredicción, y una única PU puede contener múltiples particiones de forma arbitraria para codificar de manera efectiva patrones de imagen irregulares. La TU se puede considerar como la unidad básica de transformación. La TU se puede definir independientemente de la PU; sin embargo, el tamaño de una TU puede estar limitado al tamaño de la CU a la que pertenece la TU. Esta separación de la estructura de bloques en tres conceptos diferentes puede permitir optimizar cada unidad de acuerdo con su función respectiva de la unidad, lo que puede resultar en una mejora de la eficacia de codificación.

[0013] Para fines de ilustración solamente, ciertos modos de realización divulgados en el presente documento se describen con ejemplos que incluyen solo dos capas (por ejemplo, una capa inferior tal como la capa base y una capa superior tal como la capa de mejora) de datos de vídeo. Una "capa" de datos de vídeo puede referirse, en general, a una secuencia de imágenes que tiene al menos una característica común, tal como una vista, una velocidad de trama, una resolución o similar. Por ejemplo, una capa puede incluir datos de vídeo asociados con una vista particular (por ejemplo, perspectiva) de datos de vídeo de múltiples vistas. Como otro ejemplo, una capa puede incluir datos de vídeo asociados a una capa particular de datos de vídeo ajustables a escala. Por lo tanto, esta divulgación puede referirse indistintamente a una capa y a una vista de datos de vídeo. Es decir, una vista de datos de vídeo puede denominarse una capa de datos de vídeo, y una capa de datos de vídeo puede denominarse una vista de datos de vídeo. Además, un códec de capas múltiples (también denominado codificador de vídeo de capas múltiples o codificador-decodificador de capas múltiples) puede referirse conjuntamente a un códec de vistas múltiples o un códec escalable (por ejemplo, un códec configurado para codificar y/o decodificar datos de vídeo utilizando MV-HEVC, 3D-HEVC, SHVC u otra técnica de codificación multicapa). La codificación de vídeo y la descodificación de vídeo pueden, en general, denominarse codificación de vídeo. Debe entenderse que dichos ejemplos pueden ser aplicables a configuraciones que incluyen múltiples capas base y/o de mejora. Además, para facilitar la explicación, la siguiente divulgación incluye los términos "tramas" o "bloques" con referencia a ciertos modos de realización. Sin embargo, estos términos no pretenden ser limitativos. Por ejemplo, las técnicas descritas a continuación se pueden usar con cualquier unidad de vídeo adecuada, como bloques (por ejemplo, CU, PU, TU, macrobloques, etc.), fragmentos, tramas, etc.

Normas de codificación de vídeo

[0014] Una imagen digital, como una imagen de vídeo, una imagen de TV, una imagen fija o una imagen generada por una grabadora de vídeo o un ordenador puede consistir en píxeles o muestras dispuestas en líneas horizontales y verticales. La cantidad de píxeles en una sola imagen suele ser de decenas de miles. Cada píxel contiene típicamente información de luminancia y crominancia. Sin compresión, la gran cantidad de información que se transmitirá desde un codificador de imágenes a un decodificador de imágenes haría imposible la transmisión de imágenes en tiempo real. Para reducir la cantidad de información a transmitir, se han desarrollado una serie de procedimientos de compresión diferentes, tales como las normas JPEG, MPEG y H.263.

[0015] Entre las normas de codificación de vídeo se 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 conocida como ISO/IEC MPEG-4 AVC), incluyendo sus extensiones SVC y MVC.

[0016] Además, existe una nueva norma de codificación de vídeo, concretamente la codificación de vídeo de alta eficiencia (HEVC), que está siendo desarrollada por el Equipo de colaboración conjunta en codificación de vídeo (JCT-VC) del Grupo de expertos en codificación de vídeo (VCEG) de ITU-T y el Grupo de expertos en imágenes en movimiento (MPEG) de ISO/IEC. La cita completa del borrador 10 de HEVC es el documento JCTVC-L1003, de Bross y otros, titulado "High Efficiency Video Coding (HEVC) Text Specification Draft 10", Equipo de colaboración conjunta en codificación de vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, 12ª reunión: Ginebra, Suiza, del 14 de enero de 2013 al 23 de enero de 2013. La ampliación multivista para HEVC, llamada MV-HEVC, y la ampliación escalable para HEVC, llamada SHVC, también están siendo desarrolladas por el JCT-3V (Equipo de

colaboración conjunta de ITU-T/ISO/IEC para el desarrollo de la ampliación de codificación de vídeo 3D) y JCT-VC, respectivamente.

Visión general

5 **[0017]** Las imágenes de puntos de acceso aleatorio (IRAP) pueden proporcionar puntos de acceso aleatorio para decodificar un flujo de bits. Un decodificador puede comenzar a decodificar un flujo de bits decodificando una imagen IRAP sin tener que decodificar imágenes que preceden a la imagen IRAP. En el momento de decodificar una imagen IRAP, la memoria intermedia de imagen decodificada (DPB) puede tener varias imágenes decodificadas en la memoria intermedia. Si la salida de las imágenes existentes en la DPB afectaría el rendimiento del decodificador (por ejemplo, existen demasiadas imágenes en la DPB para que el decodificador pueda emitirlas), puede ser conveniente eliminar esas imágenes existentes sin emitirlas (por ejemplo, imágenes).

15 **[0018]** El NoOutputOfPriorPicsFlag variable puede indicar, al decodificar una imagen IRAP, si las imágenes en la DPB deben enviarse antes de ser eliminadas de la DPB. Por ejemplo, al decodificar una imagen IRAP, el valor de NoOutputOfPriorPicsFlag se puede establecer en 1 cuando las imágenes en la DPB no se deben emitir antes de eliminarlas. El valor de NoOutputOfPriorPicsFlag se puede determinar en función de un elemento de sintaxis correspondiente y/o diversas condiciones e información. En versiones anteriores de HEVC, SHVC y MV-HEVC (por ejemplo, borrador de trabajo SHVC 4 y borrador de trabajo MV-HEVC 6), el NoOutputOfPriorPicsFlag puede establecerse igual a 1 cuando la resolución espacial o el número de almacenamientos de imágenes cambia. Al establecer NoOutputOfPriorPicsFlag en 1, el decodificador puede controlar la cantidad de memoria disponible en la DPB eliminando las imágenes en la DPB según corresponda. Sin embargo, la cantidad de memoria necesaria en la DPB también puede verse afectada por cambios en el formato de color y/o la profundidad de bits de las imágenes.

25 **[0019]** Para abordar estos y otros desafíos, las técnicas de acuerdo con ciertos aspectos pueden tener en cuenta el formato de color y/o la profundidad de bits de las imágenes al determinar el valor de NoOutputOfPriorPicsFlag para las imágenes IRAP. Por ejemplo, las técnicas pueden determinar si el formato de color y/o la profundidad de bits de la imagen actual en la capa actual a decodificar son diferentes del formato de color y/o la profundidad de bits de la imagen anterior en la capa actual. El formato de color y/o la profundidad de bits de la imagen actual y la imagen anterior se pueden obtener del conjunto de parámetros de secuencia actualmente activos (SPS) y el SPS que estaba activo cuando se decodificó la imagen anterior, respectivamente. Si hay cambios en el formato de color o en la profundidad de bits, el valor de NoOutputOfPriorPicsFlag puede establecerse igual a 1 para activar la eliminación de imágenes en la DPB sin generar las imágenes.

35 **[0020]** Al tener en cuenta los cambios en el formato de color y/o la profundidad de bits de las imágenes, las técnicas pueden determinar el valor de NoOutputOfPriorPicsFlag con mayor precisión. Por ejemplo, las técnicas pueden anticipar mejor situaciones que requerirían memoria adicional en la memoria intermedia.

Sistema de codificación de vídeo

40 **[0021]** De aquí en adelante, en el presente documento se describen de forma más detallada diversos aspectos de los sistemas, aparatos y procedimientos novedosos, con referencia a los dibujos adjuntos. Sin embargo, la presente divulgación se puede realizar de muchas formas diferentes y no se debería interpretar que está limitada a ninguna estructura o función específicas presentadas a lo largo de esta divulgación. En cambio, estos aspectos se proporcionan de modo que esta divulgación sea exhaustiva y completa, y transmita por completo el alcance de la divulgación a los expertos en la técnica. Basándose en las enseñanzas del presente documento, un experto en la materia debería apreciar que el alcance de la divulgación pretende abarcar cualquier aspecto de los sistemas, aparatos y procedimientos novedosos divulgados en el presente documento, ya sean implementados de forma independiente de, o en combinación con, cualquier otro aspecto de la presente divulgación. Por ejemplo, un aparato se puede implementar o un procedimiento se puede llevar a la práctica usando un número cualquiera de los aspectos expuestos en el presente documento. Además, el alcance de la presente divulgación pretende abarcar dicho aparato o procedimiento que se lleva a la práctica usando otra estructura, funcionalidad o estructura y funcionalidad, además, o aparte, de los diversos aspectos de la presente divulgación expuestos en el presente documento. Debe entenderse que cualquier aspecto divulgado en el presente documento se puede realizar mediante uno o más elementos de una reivindicación.

55 **[0022]** Aunque en el presente documento se describen unos aspectos en particular, muchas variantes y permutaciones de estos aspectos se hallan dentro del alcance de la divulgación. Aunque se mencionan algunos beneficios y ventajas de los aspectos preferentes, el alcance de la divulgación no está concebido para estar limitado a beneficios, usos u objetivos en particular. En cambio, los aspectos de la divulgación están concebidos para ser ampliamente aplicables a diferentes tecnologías inalámbricas, configuraciones de sistema, redes y protocolos de transmisión, algunos de los cuales se ilustran a modo de ejemplo en las figuras y en la siguiente descripción de los aspectos preferentes. La descripción detallada y los dibujos son meramente ilustrativos de la divulgación, en lugar de limitantes, estando definido el alcance de la divulgación por las reivindicaciones adjuntas y equivalentes de las mismas.

[0023] Los dibujos adjuntos ilustran ejemplos. Los elementos indicados mediante números de referencia en los dibujos adjuntos corresponden a elementos indicados mediante números de referencia similares en la siguiente descripción. En esta divulgación, los elementos que tienen nombres que comienzan con palabras ordinales (por ejemplo, "primero", "segundo", "tercero", etc.) no necesariamente implican que los elementos tienen un orden particular. Más bien, dichas palabras ordinales se usan simplemente para referirse a diferentes elementos de un mismo tipo o un tipo similar.

[0024] La **FIG. 1A** es un diagrama de bloques que ilustra un sistema de codificación de vídeo 10 de ejemplo que puede usar técnicas de acuerdo con aspectos descritos en esta divulgación. Como se usa en el presente documento, el término "codificador de vídeo" se refiere, en general, tanto a codificadores de vídeo como a descodificadores de vídeo. En esta divulgación, los términos "codificación de vídeo" o "codificación" pueden referirse genéricamente a la codificación de vídeo y a la descodificación de vídeo. Además de los codificadores de vídeo y los descodificadores de vídeo, los aspectos descritos en la presente solicitud pueden extenderse a otros dispositivos relacionados, tales como los transcodificadores (por ejemplo, dispositivos que pueden descodificar un flujo de bits y recodificar otro flujo de bits) y cajas intermedias (por ejemplo, dispositivos que pueden modificar, transformar y/o manipular de otro modo un flujo de bits).

[0025] Como se muestra en la **FIG. 1A**, el sistema de codificación de vídeo 10 incluye un dispositivo de origen 12 que genera datos de vídeo codificados para ser descodificados posteriormente por un dispositivo de destino 14. En el ejemplo de la **FIG. 1A**, el dispositivo de origen 12 y el dispositivo de destino 14 constituyen dispositivos independientes. Sin embargo, se observa que los dispositivos de origen y de destino 12, 14 pueden estar en el mismo dispositivo, o ser parte del mismo, como se muestra en el ejemplo de la **FIG. 1B**.

[0026] Con referencia una vez más a la **FIG. 1A**, el dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender, respectivamente, cualquiera entre una amplia gama de dispositivos, incluidos los ordenadores de escritorio, los ordenadores plegables (por ejemplo, los ordenadores portátiles), los ordenadores de tableta, los descodificadores, los equipos telefónicos tales como los llamados "teléfonos inteligentes", los llamados paneles "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivos de transmisión de flujo de vídeo o similares. En algunos casos, el dispositivo de origen 12 y el dispositivo de destino 14 pueden estar equipados para la comunicación inalámbrica.

[0027] El dispositivo de destino 14 puede recibir, a través del enlace 16, los datos de vídeo codificados a descodificar. El enlace 16 puede comprender cualquier tipo de medio o dispositivo capaz de mover los datos de vídeo codificados desde el dispositivo de origen 12 al dispositivo de destino 14. En el ejemplo de la **FIG. 1A**, el enlace 16 puede comprender un medio de comunicación para permitir que el dispositivo de origen 12 transmita datos de vídeo codificados al dispositivo de destino 14 en tiempo real. Los datos de vídeo codificados pueden modularse de acuerdo con una norma de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitirse al dispositivo de destino 14. El medio de comunicación puede comprender cualquier medio de comunicación inalámbrica o alámbrica tal como un espectro de radiofrecuencia (RF) o una o más líneas físicas de transmisión. 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 tal como Internet. El medio de comunicación puede incluir encaminadores, conmutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo de origen 12 al dispositivo de destino 14.

[0028] De forma alternativa, los datos codificados pueden emitirse desde una interfaz de salida 22 hasta un dispositivo de almacenamiento 31 optativo. De forma similar, se puede acceder a los datos codificados desde el dispositivo de almacenamiento 31 mediante la interfaz de entrada 28, por ejemplo, del dispositivo de destino 14. El dispositivo de almacenamiento 31 puede incluir cualquiera entre una diversidad de medios de almacenamiento de datos, de acceso distribuido o local, tales como una unidad de disco duro, memoria flash, memoria volátil o no volátil u otros medios adecuados cualesquiera de almacenamiento digital para almacenar datos de vídeo codificados. En otro ejemplo, el dispositivo de almacenamiento 31 puede corresponder a un servidor de archivos o a otro dispositivo de almacenamiento intermedio que pueda guardar el vídeo codificado generado mediante el dispositivo de origen 12. El dispositivo de destino 14 puede acceder a datos de vídeo almacenados del dispositivo de almacenamiento 31, mediante transmisión continua 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 de destino 14. Los ejemplos de servidores de archivos incluyen un servidor de la red (por ejemplo, para una página de la red), un servidor de protocolo de transferencia de archivos (FTP), dispositivos de almacenamiento anexos a la red (NAS) o una unidad de disco local. El dispositivo de destino 14 puede acceder a los datos de vídeo codificados mediante cualquier conexión de datos estándar, incluyendo una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión inalámbrica de red de área local (WLAN)), una conexión cableada (por ejemplo, una línea de abonado digital (DSL), un módem de cable, etc.), o una combinación de ambos que sea adecuada para acceder a los datos de vídeo codificados almacenados en un servidor de archivos. La transmisión de datos de vídeo codificados desde el dispositivo de almacenamiento 31 puede ser una transmisión continua, una transmisión de descarga o una combinación de ambas.

[0029] Las técnicas de esta divulgación no están limitadas a aplicaciones o configuraciones inalámbricas. Las técnicas pueden aplicarse a la codificación de vídeo, para admitir cualquiera de una diversidad de aplicaciones multimedia, tales como radiodifusiones de televisión por el aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de vídeo en continuo, por ejemplo, a través de Internet (por ejemplo, transmisión adaptativa dinámica sobre protocolo de transferencia de hipertexto (HTTP), etc.), codificación de vídeo digital para su almacenamiento en un medio de almacenamiento de datos, descodificación de vídeo digital almacenado en un medio de almacenamiento de datos, u otras aplicaciones. En algunos ejemplos, el sistema de codificación de vídeo 10 puede estar configurado para admitir transmisión de vídeo unidireccional o bidireccional para admitir aplicaciones tales como la transmisión continua de vídeo, la reproducción de vídeo, la radiodifusión de vídeo y/o la videotelefonía.

[0030] En el ejemplo de la **FIG. 1A**, el dispositivo de origen 12 incluye un origen de vídeo 18, un codificador de vídeo 20 y la interfaz de salida 22. En algunos casos, la interfaz de salida 22 puede incluir un modulador/demodulador (módem) y/o un transmisor. En el dispositivo de origen 12, el origen de vídeo 18 puede incluir un origen tal como un dispositivo de captura de vídeo, por ejemplo, una videocámara, un archivo de vídeo que contiene vídeo previamente capturado, una interfaz de alimentación de vídeo para recibir vídeo desde un proveedor de contenido de vídeo y/o un sistema de gráficos por ordenador para generar datos de gráficos por ordenador como el vídeo de origen, o una combinación de dichos orígenes. Como un ejemplo, si el origen de vídeo 18 es una videocámara, el dispositivo de origen 12 y el dispositivo de destino 14 pueden formar los denominados "teléfonos de cámara" o "videoteléfonos", tal como se ilustra en el ejemplo de la **FIG. 1B**. Sin embargo, las técnicas descritas en esta divulgación pueden ser aplicables a la codificación de vídeo en general, y se pueden aplicar a aplicaciones inalámbricas y/o cableadas.

[0031] El vídeo capturado, precapturado o generado por ordenador puede ser codificado por el codificador de vídeo 20. Los datos de vídeo codificados pueden ser transmitidos al dispositivo de destino 14 mediante la interfaz de salida 22 del dispositivo de origen 12. Los datos de vídeo codificados pueden almacenarse también (o de forma alternativa) en el dispositivo de almacenamiento 31 para un acceso posterior mediante el dispositivo de destino 14 u otros dispositivos, para su descodificación y/o reproducción. El codificador de vídeo 20 ilustrado en la **FIG. 1A** y la **FIG. 1B** puede comprender el codificador de vídeo 20 ilustrado en la **FIG. 2A**, el codificador de vídeo 23 mostrado en la **FIG. 2B**, o cualquier otro codificador de vídeo descrito en el presente documento.

[0032] En el ejemplo de la **FIG. 1A**, el módulo de destino 14 incluye una interfaz de entrada 28, un descodificador de vídeo 30 y un dispositivo de visualización 32. En algunos casos, la interfaz de entrada 28 puede incluir un receptor y/o un módem. La interfaz de entrada 28 del dispositivo de destino 14 puede recibir los datos de vídeo codificados por el enlace 16 y/o desde el dispositivo de almacenamiento 31. Los datos de vídeo codificados, comunicados a través del enlace 16, o proporcionados en el dispositivo de almacenamiento 31, pueden incluir una diversidad de elementos sintácticos generados mediante el codificador de vídeo 20, para su uso mediante un decodificador de vídeo, tal como el decodificador de vídeo 30, en la decodificación de los datos de vídeo. Dichos elementos sintácticos se pueden incluir con los datos de vídeo codificados, transmitidos en un medio de comunicación, almacenados en un medio de almacenamiento o almacenados en un servidor de archivos. El descodificador de vídeo 30 ilustrado en la **FIG. 1A** y la **FIG. 1B** puede comprender el descodificador de vídeo 30 mostrado en la **FIG. 3A**, el descodificador de vídeo 33 mostrado en la **FIG. 3B**, o cualquier otro descodificador de vídeo descrito en el presente documento.

[0033] El dispositivo de visualización 32 puede estar integrado con, o ser externo a, el dispositivo de destino 14. En algunos ejemplos, el dispositivo de destino 14 puede incluir un dispositivo de visualización integrado y también estar configurado para interconectarse con un dispositivo de visualización externo. En otros ejemplos, el dispositivo de destino 14 puede ser un dispositivo de visualización. En general, el dispositivo de visualización 32 muestra los datos de vídeo decodificados a un usuario, y puede comprender cualquiera entre varios dispositivos de visualización, tales como una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodos orgánicos emisores de luz (OLED) u otro tipo de dispositivo de visualización.

[0034] En aspectos relacionados, la **FIG. 1B** muestra un sistema de codificación y descodificación de vídeo 10' de ejemplo en el que los dispositivos de origen y de destino 12, 14 están en, o forman parte de, un dispositivo 11. El dispositivo 11 puede ser un equipo telefónico de mano, tal como un teléfono "inteligente" o similar. El dispositivo 11 puede incluir un dispositivo controlador/procesador 13 optativo en comunicación operativa con los dispositivos de origen y de destino 12, 14. El sistema 10' de la **FIG. 1B** puede incluir, además, una unidad de procesamiento de vídeo 21 entre el codificador de vídeo 20 y la interfaz de salida 22. En algunas implementaciones, la unidad de procesamiento de vídeo 21 es una unidad separada, como se ilustra en la **FIG. 1B**; sin embargo, en otras implementaciones, la unidad de procesamiento de vídeo 21 puede implementarse como una porción del codificador de vídeo 20 y/o el dispositivo procesador/controlador 13. El sistema 10' también puede incluir un rastreador 29 optativo, que puede rastrear un objeto de interés en una secuencia de vídeo. El objeto o interés a rastrear puede ser segmentado por una técnica descrita en relación con uno o más aspectos de la presente divulgación. En aspectos relacionados, el rastreo puede ser realizado por el dispositivo de visualización 32, solo o junto con el rastreador 29. El sistema 10' de la **FIG. 1B**, y sus componentes, son similares de otro modo al sistema 10 de la **FIG. 1A** y sus componentes.

[0035] El codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con una norma de compresión de vídeo, tal como el HEVC, y pueden ajustarse a un modelo de prueba HEVC (HM). De forma alternativa,

el codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con otras normas privadas o de la industria, tales como la norma ITU-T H.264, denominada de forma alternativa MPEG-4, Parte 10, AVC, o ampliaciones de dichas normas. Sin embargo, las técnicas de esta divulgación no están limitadas a ninguna norma de codificación particular. Otros ejemplos de normas de compresión de vídeo incluyen MPEG-2 e ITU-T H.263.

[0036] Aunque no se muestra en los ejemplos de la **FIG. 1A** y la **FIG. 1B**, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden estar integrados, cada uno, con un codificador y descodificador de audio, y pueden incluir unidades MUX-DEMUX apropiadas, u otro hardware y software, para gestionar la codificación tanto de audio como de vídeo en un flujo de datos común o flujos de datos por separado. Si procede, en algunos ejemplos, las unidades MUX-DEMUX pueden ajustarse al protocolo de multiplexor ITU H.223 o a otros protocolos, tales como el protocolo de datagramas de usuario (UDP).

[0037] Cada uno entre el codificador de vídeo 20 y el decodificador de vídeo 30 puede implementarse como cualquiera entre una variedad de circuitos codificadores adecuados, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), formaciones de compuertas programables in situ (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 entre el codificador de vídeo 20 y el decodificador de vídeo 30 puede estar incluido en uno o más codificadores o decodificadores, cualquiera de los cuales puede estar integrado como parte de un codificador/decodificador (CÓDEC) combinado en un respectivo dispositivo.

Proceso de codificación de vídeo

[0038] Como se ha mencionado brevemente en lo que antecede, el codificador de vídeo 20 codifica datos de vídeo. Los datos de vídeo pueden comprender una o más imágenes. Cada una de las imágenes es una imagen fija que forma parte de un vídeo. En algunos casos, una imagen puede denominarse una "trama" de vídeo. Cuando el codificador de vídeo 20 codifica los datos de vídeo, el codificador de vídeo 20 puede generar un flujo de bits. El flujo de bits puede incluir una secuencia de bits que forman una representación codificada de los datos de vídeo. El flujo de bits puede incluir imágenes codificadas y datos asociados. Una imagen codificada es una representación codificada de una imagen.

[0039] Para generar el flujo de bits, el codificador de vídeo 20 puede llevar a cabo operaciones de codificación en cada imagen en los datos de vídeo. Cuando el codificador de vídeo 20 realiza operaciones de codificación en las imágenes, el codificador de vídeo 20 puede generar una serie de imágenes codificadas y datos asociados. Los datos asociados pueden incluir conjuntos de parámetros de vídeo (VPS), conjuntos de parámetros de secuencia (SPS), conjuntos de parámetros de imagen (PPS), conjuntos de parámetros de adaptación (APS) y otras estructuras sintácticas. Un SPS puede contener parámetros aplicables a cero o más secuencias de imágenes. Un PPS puede contener parámetros aplicables a cero o más imágenes. Un APS puede contener parámetros aplicables a cero o más imágenes. Los parámetros de un APS pueden ser parámetros que tienen más probabilidades de cambiar que los parámetros de un PPS.

[0040] Para generar una imagen codificada, el codificador de vídeo 20 puede dividir una imagen en bloques de vídeo de igual tamaño. Un bloque de vídeo puede ser una matriz bidimensional de muestras. Cada uno de los bloques de vídeo está asociado a un bloque de árbol. En algunos casos, un bloque de árbol también puede denominarse unidad de codificación más grande (LCU). Los bloques de árbol de HEVC pueden ser aproximadamente análogos a los macrobloques de normas anteriores, tales como la H.264/AVC. Sin embargo, un bloque de árbol no está limitado necesariamente a un tamaño particular y puede incluir una o más unidades de codificación (CU). El codificador de vídeo 20 puede usar la división en árbol cuaternario para dividir los bloques de vídeo de bloques de árbol en bloques de vídeo asociados con las CU, de ahí el nombre "bloques de árbol".

[0041] En algunos ejemplos, el codificador de vídeo 20 puede dividir una imagen en una pluralidad de fragmentos. Cada uno de los fragmentos puede incluir un número entero de CU. En algunos casos, un fragmento comprende un número entero de bloques de árbol. En otros casos, un límite de un fragmento puede estar dentro de un bloque de árbol.

[0042] Como parte del modo de realización de una operación de codificación en una imagen, el codificador de vídeo 20 puede realizar operaciones de codificación en cada fragmento de la imagen. Cuando el codificador de vídeo 20 realiza una operación de codificación en un fragmento, el codificador de vídeo 20 puede generar datos codificados asociados con el fragmento. Los datos codificados asociados al fragmento pueden denominarse "fragmento codificado".

[0043] Para generar un fragmento codificado, el codificador de vídeo 20 puede llevar a cabo operaciones de codificación en cada bloque de árbol en un fragmento. Cuando el codificador de vídeo 20 realiza una operación de

codificación en un bloque de árbol, el codificador de vídeo 20 puede generar un bloque de árbol codificado. El bloque de árbol codificado puede comprender datos que representan una versión codificada del bloque de árbol.

5 **[0044]** Cuando el codificador de vídeo 20 genera un fragmento codificado, el codificador de vídeo 20 puede realizar operaciones de codificación (por ejemplo, codificar) en los bloques de árbol en el fragmento de acuerdo con una orden de escaneo de trama. Por ejemplo, el codificador de vídeo 20 puede codificar los bloques de árbol del fragmento en un orden que avanza de izquierda a derecha en la fila más alta de los bloques de árbol en el fragmento, luego de izquierda a derecha en la siguiente fila inferior de los bloques de árbol, y así sucesivamente hasta que el codificador de vídeo 20 ha codificado cada uno de los bloques de árbol del fragmento.

10 **[0045]** Como resultado de codificar los bloques de árbol de acuerdo con el orden de escaneo de trama, los bloques de árbol de arriba y a la izquierda de un bloque de árbol dado pueden haber sido codificados, pero los bloques de árbol de debajo y a la derecha del bloque de árbol dado aún no han sido codificados. En consecuencia, el codificador de vídeo 20 puede tener acceso a la información generada por la codificación de los bloques de árbol de arriba y a la izquierda del bloque de árbol determinado cuando codifica el bloque de árbol determinado. Sin embargo, el codificador de vídeo 20 puede no ser capaz de acceder a la información generada por la codificación de los bloques de árbol de debajo y a la derecha del bloque de árbol determinado al codificar el bloque de árbol determinado.

15 **[0046]** Para generar un bloque de árbol codificado, el codificador de vídeo 20 puede llevar a cabo de forma recursiva la división en árbol cuaternario en el bloque de vídeo del bloque de árbol para dividir el bloque de vídeo en bloques de vídeo progresivamente más pequeños. Cada uno de los bloques de vídeo más pequeños puede estar asociado a una CU diferente. Por ejemplo, el codificador de vídeo 20 puede dividir el bloque de vídeo de un bloque de árbol en cuatro subbloques de igual tamaño, dividir uno o más de los subbloques en cuatro subsubbloques de igual tamaño, etc. Una CU dividida puede ser una CU cuyo bloque de vídeo está dividido en bloques de vídeo asociados a otras CU. Una CU no dividida puede ser una CU cuyo bloque de vídeo no esté dividido en bloques de vídeo asociados a otras CU.

20 **[0047]** Uno o más elementos sintácticos en el flujo de bits pueden indicar un número máximo de veces que el codificador de vídeo 20 puede dividir el bloque de vídeo de un bloque de árbol. Un bloque de vídeo de una CU puede tener forma cuadrada. El tamaño del bloque de vídeo de una CU (es decir, el tamaño de la CU) puede variar desde 8x8 píxeles hasta el tamaño de un bloque de vídeo de un bloque de árbol (es decir, el tamaño del bloque de árbol) con un máximo de 64x64 píxeles o mayor.

25 **[0048]** El codificador de vídeo 20 puede realizar operaciones de codificación (por ejemplo, codificar) en cada CU de un bloque de árbol de acuerdo con un orden de escaneo z. En otras palabras, el codificador de vídeo 20 puede codificar una CU superior izquierda, una CU superior derecha, una CU inferior izquierda y luego una CU inferior derecha, en ese orden. Cuando el codificador de vídeo 20 realiza una operación de codificación en una CU dividida, el codificador de vídeo 20 puede codificar las CU asociadas con subbloques del bloque de vídeo de la CU dividida de acuerdo con el orden de escaneo z. En otras palabras, el codificador de vídeo 20 puede codificar una CU asociada con un subbloque superior izquierdo, una CU asociada con un subbloque superior derecho, una CU asociada con un subbloque inferior izquierdo, y luego una CU asociada con un subbloque inferior derecho, en ese orden.

30 **[0049]** Como resultado de la codificación de las CU de un bloque de árbol de acuerdo con un orden de escaneo en z, las CU de arriba, de arriba y a la izquierda, de arriba y a la derecha, a la izquierda y de abajo y a la izquierda de una CU dada pueden haber sido codificadas. Las CU de abajo y a la derecha de la CU determinada todavía no han sido codificadas. En consecuencia, el codificador de vídeo 20 puede ser capaz de acceder a la información generada codificando algunas CU que se encuentran próximas a la CU determinada cuando codifica la CU determinada. Sin embargo, el codificador de vídeo 20 puede ser incapaz de acceder a la información generada por la codificación de otras CU que se encuentran junto a la CU determinada cuando codifica la CU determinada.

35 **[0050]** Cuando el codificador de vídeo 20 codifica una CU no dividida, el codificador de vídeo 20 puede generar una o más unidades de predicción (PU) para la CU. Cada una de las PU de la CU puede estar asociada a un bloque de vídeo diferente dentro del bloque de vídeo de la CU. El codificador de vídeo 20 puede generar un bloque de vídeo predicho para cada PU de la CU. El bloque de vídeo predicho de una PU puede ser un bloque de muestras. El codificador de vídeo 20 puede usar intrapredicción o interpredicción para generar el bloque de vídeo predicho para una PU.

40 **[0051]** Cuando el codificador de vídeo 20 utiliza la intrapredicción para generar el bloque de vídeo predicho de una PU, el codificador de vídeo 20 puede generar el bloque de vídeo predicho de la PU basándose en muestras descodificadas de la imagen asociada con la PU. Si el codificador de vídeo 20 utiliza la intrapredicción para generar bloques de vídeo predichos de las PU de una CU, la CU es una CU intrapredicha. Si el codificador de vídeo 20 utiliza la interpredicción para generar el bloque de vídeo predicho de la PU, el codificador de vídeo 20 puede generar el bloque de vídeo predicho de la PU basándose en muestras descodificadas de una o más imágenes distintas a la imagen asociada con la PU. Si el codificador de vídeo 20 utiliza la interpredicción para generar bloques de vídeo predichos de las PU de una CU, la CU es una CU interpredicha.

- 5 **[0052]** Además, cuando el codificador de vídeo 20 utiliza la interpredicción para generar un bloque de vídeo predicho para una PU, el codificador de vídeo 20 puede generar información de movimiento para la PU. La información de movimiento para una PU puede indicar uno o más bloques de referencia de la PU. Cada bloque de referencia de la PU puede ser un bloque de vídeo dentro de una imagen de referencia. La imagen de referencia puede ser una imagen distinta de la imagen asociada a la PU. En algunos casos, un bloque de referencia de una PU también puede denominarse "muestra de referencia" de la PU. El codificador de vídeo 20 puede generar el bloque de vídeo predicho para la PU en base a los bloques de referencia de la PU.
- 10 **[0053]** Después de que el codificador de vídeo 20 genere bloques de vídeo predichos para una o más PU de una CU, el codificador de vídeo 20 puede generar datos residuales para la CU basándose en los bloques de vídeo predichos para las PU de la CU. Los datos residuales para la CU pueden indicar diferencias entre muestras en los bloques de vídeo predichos para las PU de la CU y el bloque de vídeo original de la CU.
- 15 **[0054]** Además, como parte del modo de realización de una operación de codificación en una CU no dividida, el codificador de vídeo 20 puede realizar la división recursiva en árbol cuaternario en los datos residuales de la CU para dividir los datos residuales de la CU en uno o más bloques de datos residuales (es decir, bloques de vídeo residuales) asociados con las unidades de transformación (TU) de la CU. Cada TU de una CU puede estar asociada a un bloque de vídeo residual diferente.
- 20 **[0055]** El codificador de vídeo 20 puede aplicar una o más transformaciones a bloques de vídeo residuales asociados con las TU para generar bloques de coeficientes de transformación (por ejemplo, bloques de coeficientes de transformación) asociados con las TU. Conceptualmente, un bloque de coeficientes de transformación puede ser una matriz bidimensional (2D) de coeficientes de transformación.
- 25 **[0056]** Después de generar un bloque de coeficientes de transformación, el codificador de vídeo 20 puede realizar un proceso de cuantificación en el bloque de coeficientes de transformación. La cuantificación se refiere, en general, a un proceso en el que coeficientes de transformada se cuantifican para reducir, posiblemente, la cantidad de datos usados para representar los coeficientes de transformada, proporcionando más compresión. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos, o a la totalidad, de los coeficientes de transformada. Por ejemplo, un coeficiente de transformada de n bits puede redondearse a la baja hasta un coeficiente de transformada de m bits durante la cuantificación, donde n es mayor que m .
- 30 **[0057]** El codificador de vídeo 20 puede asociar cada CU con un valor de parámetro de cuantificación (QP). El valor QP asociado con una CU puede determinar cómo el codificador de vídeo 20 cuantifica los bloques de coeficientes de transformación asociados con la CU. El codificador de vídeo 20 puede ajustar el grado de cuantificación aplicado a los bloques de coeficientes de transformación asociados con una CU ajustando el valor QP asociado con la CU.
- 35 **[0058]** Después de que el codificador de vídeo 20 cuantifica un bloque de coeficientes de transformación, el codificador de vídeo 20 puede generar conjuntos de elementos sintácticos que representan los coeficientes de transformación en el bloque de coeficientes de transformación cuantificado. El codificador de vídeo 20 puede aplicar operaciones de codificación por entropía, tales como operaciones de codificación aritmética binaria adaptativa al contexto (CABAC), a algunos de estos elementos sintácticos. También podrían usarse otras técnicas de codificación por entropía, tales como la codificación de longitud variable adaptable al contexto (CAVLC), la codificación por entropía de división de intervalo de probabilidad (PIPE) u otra codificación aritmética binaria.
- 40 **[0059]** El flujo de bits generado por el codificador de vídeo 20 puede incluir una serie de unidades de capa de abstracción de red (NAL). Cada una de las unidades NAL puede ser una estructura sintáctica que contiene una indicación de un tipo de datos en la unidad NAL y octetos que contienen los datos. Por ejemplo, una unidad NAL puede contener datos que representan un conjunto de parámetros de vídeo, un conjunto de parámetros de secuencia, un conjunto de parámetros de imagen, un fragmento codificado, información de mejora complementaria (SEI), un delimitador de unidad de acceso, datos de relleno u otro tipo de datos. Los datos de una unidad NAL pueden incluir varias estructuras sintácticas.
- 45 **[0060]** El decodificador de vídeo 30 puede recibir el flujo de bits generado por el codificador de vídeo 20. El flujo de bits puede incluir una representación codificada de los datos de vídeo codificados por el codificador de vídeo 20. Cuando el decodificador de vídeo 30 recibe el flujo de bits, el decodificador de vídeo 30 puede realizar una operación de análisis sintáctico en el flujo de bits. Cuando el decodificador de vídeo 30 realiza la operación de análisis sintáctico, el decodificador de vídeo 30 puede extraer elementos sintácticos del flujo de bits. El decodificador de vídeo 30 puede reconstruir las imágenes de los datos de vídeo basándose en los elementos sintácticos extraídos del flujo de bits. El proceso para reconstruir los datos de vídeo basados en los elementos sintácticos puede ser, en general, recíproco al proceso realizado por el codificador de vídeo 20 para generar los elementos sintácticos.
- 50 **[0061]** Después de que el decodificador de vídeo 30 extraiga los elementos sintácticos asociados con una CU, el decodificador de vídeo 30 puede generar bloques de vídeo predichos para las PU de la CU basándose en los elementos sintácticos. Además, el decodificador de vídeo 30 puede cuantificar de forma inversa bloques de coeficientes de transformación asociados con las TU de la CU. El decodificador de vídeo 30 puede realizar
- 55
- 60
- 65

transformaciones inversas en los bloques de coeficientes de transformación para reconstruir los bloques de vídeo residuales asociados con las TU de la CU. Después de generar los bloques de vídeo predichos y reconstruir los bloques de vídeo residuales, el descodificador de vídeo 30 puede reconstruir el bloque de vídeo de la CU basándose en los bloques de vídeo predichos y los bloques de vídeo residuales. De esta manera, el descodificador de vídeo 30 puede reconstruir los bloques de vídeo de las CU basándose en los elementos sintácticos del flujo de bits.

Codificador de vídeo

[0062] La FIG. 2A es un diagrama de bloques que ilustra un ejemplo de un codificador de vídeo 20 que puede implementar técnicas de acuerdo a aspectos descritos en esta divulgación. El codificador de vídeo 20 puede configurarse para procesar una sola capa de una trama de vídeo, tal como para HEVC. Además, el codificador de vídeo 20 puede configurarse para realizar cualquiera o todas las técnicas de esta divulgación, que incluyen, pero no se limitan a, los procedimientos para inferir NoOutputOfPriorPicsFlag y procesos relacionados descritos con mayor detalle anteriormente y posteriormente con respecto a las FIGS. 4 y 5. Como ejemplo, la unidad de procesamiento de predicción 100 puede configurarse para realizar cualquiera o todas las técnicas descritas en esta divulgación. En otro modo de realización, el codificador de vídeo 20 incluye una unidad 128 de predicción entre capas opcional que está configurada para realizar cualquiera o todas las técnicas descritas en esta divulgación. En otros modos de realización, la predicción entre capas se puede realizar mediante la unidad de procesamiento de predicción 100 (por ejemplo, la unidad de interpredicción 121 y/o la unidad de intrapredicción 126), en cuyo caso la unidad 128 de predicción entre capas puede omitirse. Sin embargo, los aspectos de esta divulgación no se limitan a lo anterior. En algunos ejemplos, las técnicas descritas en esta divulgación pueden compartirse entre los diversos componentes del codificador de vídeo 20. En algunos ejemplos, adicionalmente o de forma alternativa, un procesador (no mostrado) puede configurarse para realizar cualquiera o todas las técnicas descritas en esta divulgación.

[0063] Con fines de explicación, esta divulgación describe el codificador de vídeo 20 en el contexto de la codificación HEVC. Sin embargo, las técnicas de esta divulgación pueden ser aplicables a otras normas o procedimientos de codificación. El ejemplo representado en la FIG. 2A es para un códec de una sola capa. Sin embargo, como se describirá adicionalmente con respecto a la FIG. 2B, algunos o todos los codificadores de vídeo 20 pueden duplicarse para procesar un códec multicapa.

[0064] El codificador de vídeo 20 puede realizar la intracodificación y la intercodificación de bloques de vídeo dentro de fragmentos de vídeo. La intracodificación se basa en la predicción espacial para reducir o eliminar la redundancia espacial en el vídeo dentro de una trama o imagen de vídeo determinada. La intercodificación se basa en la predicción temporal para reducir o eliminar la redundancia temporal de un vídeo dentro de tramas o imágenes adyacentes de una secuencia de vídeo. Intramodo (modo I) se puede referir a cualquiera de varios modos de codificación de base espacial. Los intermodos, tales como la predicción unidireccional (modo P) o la predicción bidireccional (modo B), pueden referirse a cualquiera de varios modos de codificación de base temporal.

[0065] En el ejemplo de la FIG. 2A, el codificador de vídeo 20 incluye una pluralidad de componentes funcionales. Los componentes funcionales del codificador de vídeo 20 incluyen una unidad de procesamiento de predicción 100, una unidad de generación residual 102, una unidad de procesamiento de transformada 104, una unidad de cuantificación 106, una unidad de cuantificación inversa 108, una unidad de transformada inversa 110, una unidad de reconstrucción 112, una unidad de filtro 113, una memoria intermedia de imágenes descodificadas 114 y una unidad de codificación por entropía 116. La unidad de procesamiento de predicción 100 incluye una unidad de interpredicción 121, una unidad de estimación de movimiento 122, una unidad de compensación de movimiento 124, una unidad de intrapredicción 126 y una unidad de predicción entre capas 128. En otros ejemplos, el codificador de vídeo 20 puede incluir más, menos o diferentes componentes funcionales. Además, la unidad de estimación de movimiento 122 y la unidad de compensación de movimiento 124 pueden estar altamente integradas, pero están representadas en el ejemplo de la FIG. 2A de forma separada con fines explicativos.

[0066] El codificador de vídeo 20 puede recibir datos de vídeo. El codificador de vídeo 20 puede recibir los datos de vídeo de diversas fuentes. Por ejemplo, el codificador de vídeo 20 puede recibir los datos de vídeo de la fuente de vídeo 18 (por ejemplo, mostrada en la FIG. 1A o 1B) u otra fuente. Los datos de vídeo pueden representar una serie de imágenes. Para codificar los datos de vídeo, el codificador de vídeo 20 puede realizar una operación de codificación en cada una de las imágenes. Como parte de realizar la operación de codificación en una imagen, el codificador de vídeo 20 puede realizar operaciones de codificación en cada fragmento de la imagen. Como parte de realizar una operación de codificación en un fragmento, el codificador de vídeo 20 puede realizar operaciones de codificación en bloques de árbol en el fragmento.

[0067] Como parte de realizar una operación de codificación en un bloque de árbol, la unidad de procesamiento de predicción 100 puede realizar la división en árbol cuaternario en el bloque de vídeo del bloque de árbol para dividir el bloque de vídeo en bloques de vídeo progresivamente más pequeños. Cada uno de los bloques de vídeo más pequeños puede estar asociado a una CU diferente. Por ejemplo, la unidad de procesamiento de predicción 100 puede dividir un bloque de vídeo de un bloque de árbol en cuatro subbloques de igual tamaño, dividir uno o más de los subbloques en cuatro subsubbloques de igual tamaño, etc.

[0068] Los tamaños de los bloques de vídeo asociados a las CU pueden variar desde muestras de 8x8 hasta el tamaño del bloque de árbol, con un máximo de muestras de 64x64 o mayores. En esta divulgación, "NxN" y "N por N" pueden usarse indistintamente para hacer referencia a las dimensiones de muestras de un bloque de vídeo en lo que respecta a dimensiones verticales y horizontales, por ejemplo, muestras de 16x16 o muestras de 16 por 16. En general, un bloque de vídeo de 16x16 tiene dieciséis muestras en una dirección vertical ($y = 16$) y dieciséis muestras en una dirección horizontal ($x = 16$). Asimismo, un bloque NxN presenta, en general, N muestras en una dirección vertical y N muestras en una dirección horizontal, donde N representa un valor entero no negativo.

[0069] Además, como parte de realizar la operación de codificación en un bloque de árbol, la unidad de procesamiento de predicción 100 puede generar una estructura de datos jerárquica en árbol cuaternario para el bloque de árbol. Por ejemplo, un bloque de árbol puede corresponder a un nodo raíz de la estructura de datos en árbol cuaternario. Si la unidad de procesamiento de predicción 100 divide el bloque de vídeo del bloque de árbol en cuatro subbloques, el nodo raíz tiene cuatro nodos secundarios en la estructura de datos en árbol cuaternario. Cada uno de los nodos secundarios corresponde a una CU asociada con uno de los subbloques. Si la unidad de procesamiento de predicción 100 divide uno de los subbloques en cuatro subsubbloques, el nodo correspondiente a la CU asociada con el subbloque puede tener cuatro nodos secundarios, cada uno de los cuales corresponde a una CU asociada con uno de los subsubbloques.

[0070] Cada nodo de la estructura de datos en árbol cuaternario puede contener datos sintácticos (por ejemplo, elementos sintácticos) para el bloque de árbol o CU correspondiente. Por ejemplo, un nodo en el árbol cuaternario puede incluir un indicador de división, que indica si el bloque de vídeo de la CU correspondiente al nodo está dividido (es decir, partido) en cuatro subbloques. Los elementos sintácticos para una CU pueden definirse de manera recursiva y pueden depender de si el bloque de vídeo de la CU está dividido en subbloques. Una CU cuyo bloque de vídeo no está dividido puede corresponder a un nodo hoja en la estructura de datos en árbol cuaternario. Un bloque de árbol codificado puede incluir datos basados en la estructura de datos en árbol cuaternario para un bloque de árbol correspondiente.

[0071] El codificador de vídeo 20 puede realizar operaciones de codificación en cada CU no dividida de un bloque de árbol. Cuando el codificador de vídeo 20 realiza una operación de codificación en una CU no dividida, el codificador de vídeo 20 genera datos que representan una representación codificada de la CU no dividida.

[0072] Como parte de realizar una operación de codificación en una CU, la unidad de procesamiento de predicción 100 puede dividir el bloque de vídeo de la CU entre una o más PU de la CU. El codificador de vídeo 20 y el descodificador de vídeo 30 pueden admitir varios tamaños de PU. Suponiendo que el tamaño de una CU particular sea $2N \times 2N$, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden admitir tamaños de PU de $2N \times 2N$ o $N \times N$ e interpredicción en tamaños de PU simétricas de $2N \times 2N$, $2N \times N$, $N \times 2N$, $N \times N$, $2N \times nU$, $nL \times 2N$, $nR \times 2N$ o similares. El codificador de vídeo 20 y el descodificador de vídeo 30 también pueden admitir divisiones asimétricas para tamaños de PU de $2N \times nU$, $2N \times nD$, $nL \times 2N$ y $nR \times 2N$. En algunos ejemplos, la unidad de procesamiento de predicción 100 puede realizar una división geométrica para dividir el bloque de vídeo de una CU entre las PU de la CU a lo largo de un límite que no coincide con los lados del bloque de vídeo de la CU en ángulos rectos.

[0073] La unidad de interpredicción 121 puede realizar la interpredicción en cada PU de la CU. La interpredicción puede proporcionar compresión temporal. Para realizar la interpredicción en una PU, la unidad de estimación de movimiento 122 puede generar información de movimiento para la PU. La unidad de compensación de movimiento 124 puede generar un bloque de vídeo predicho para la PU basándose en la información de movimiento y las muestras descodificadas de imágenes distintas a la imagen asociada a la CU (por ejemplo, imágenes de referencia). En esta divulgación, un bloque de vídeo predicho generado por la unidad de compensación de movimiento 124 se puede denominar un bloque de vídeo interpredicho.

[0074] Los fragmentos pueden ser fragmentos I, fragmentos P o fragmentos B. El módulo de estimación de movimiento 122 y la unidad de compensación de movimiento 124 pueden realizar diferentes operaciones para una PU de una CU dependiendo de si la PU está en un fragmento I, un fragmento P o un fragmento B. En un fragmento I, todas las PU se intrapredicen. Por lo tanto, si la PU está en un fragmento I, la unidad de estimación de movimiento 122 y la unidad de compensación de movimiento 124 no realizan la interpredicción en la PU.

[0075] Si la PU está en un fragmento P, la imagen que contiene la PU está asociada a una lista de imágenes de referencia denominada "lista 0". Cada una de las imágenes de referencia de la lista 0 contiene muestras que pueden usarse para la interpredicción de otras imágenes. Cuando la unidad de estimación de movimiento 122 realiza la operación de estimación de movimiento con respecto a una PU en un fragmento P, la unidad de estimación de movimiento 122 puede buscar las imágenes de referencia en la lista 0 para un bloque de referencia para la PU. El bloque de referencia de la PU puede ser un conjunto de muestras, por ejemplo, un bloque de muestras, que se corresponde más estrechamente con las muestras del bloque de vídeo de la PU. La unidad de estimación de movimiento 122 puede usar una variedad de métricas para determinar cuán estrechamente un conjunto de muestras en una imagen de referencia se corresponde con las muestras del bloque de vídeo de una PU. Por ejemplo, la unidad de estimación de movimiento 122 puede determinar cuán estrechamente un conjunto de muestras en una imagen de

referencia se corresponde con las muestras en el bloque de vídeo de una PU por la suma de la diferencia absoluta (SAD), la suma de diferencias al cuadrado (SSD) u otras métricas de diferencia.

[0076] Después de identificar un bloque de referencia de una PU en un fragmento P, la unidad de estimación de movimiento 122 puede generar un índice de referencia que indica la imagen de referencia de la lista 0 que contiene el bloque de referencia y un vector de movimiento que indica un desplazamiento espacial entre la PU y el bloque de referencia. En varios ejemplos, la unidad de estimación de movimiento 122 puede generar vectores de movimiento con diferentes grados de precisión. Por ejemplo, la unidad de estimación de movimiento 122 puede generar vectores de movimiento con una precisión de un cuarto de muestra, una precisión de un octavo de muestra o una precisión de otra fracción de muestra. En el caso de la precisión de fracción de muestra, los valores de bloque de referencia pueden interpolarse a partir de valores enteros de posición de muestra en la imagen de referencia. La unidad de estimación de movimiento 122 puede proporcionar el índice de referencia y el vector de movimiento como información de movimiento de la PU. La unidad de compensación de movimiento 124 puede generar el bloque de vídeo predicho de la PU basándose en el bloque de referencia identificado por la información de movimiento de la PU.

[0077] Si la PU está en un fragmento B, la imagen que contiene la PU puede estar asociada a dos listas de imágenes de referencia, denominadas "lista 0" y "lista 1". En algunos ejemplos, una imagen que contiene un fragmento B puede estar asociada a una combinación de listas que es una combinación de la lista 0 y la lista 1.

[0078] Además, si la PU está en un fragmento B, la unidad de estimación de movimiento 122 puede llevar a cabo la predicción unidireccional o la predicción bidireccional para la PU. Cuando la unidad de estimación de movimiento 122 realiza la predicción unidireccional para la PU, la unidad de estimación de movimiento 122 puede buscar las imágenes de referencia de la lista 0 o la lista 1 para un bloque de referencia para la PU. La unidad de estimación de movimiento 122 puede entonces generar un índice de referencia que indica la imagen de referencia en la lista 0 o la lista 1 que contiene el bloque de referencia y un vector de movimiento que indica un desplazamiento espacial entre la PU y el bloque de referencia. La unidad de estimación de movimiento 122 puede emitir el índice de referencia, un indicador de la dirección de predicción y el vector de movimiento como la información de movimiento de la PU. El indicador de la dirección de predicción puede indicar si el índice de referencia indica una imagen de referencia en la lista 0 o la lista 1. La unidad de compensación de movimiento 124 puede generar el bloque de vídeo predicho de la PU basándose en el bloque de referencia indicado por la información de movimiento de la PU.

[0079] Cuando la unidad de estimación de movimiento 122 realiza la predicción bidireccional para una PU, la unidad de estimación de movimiento 122 puede buscar las imágenes de referencia de la lista 0 para un bloque de referencia para la PU y también puede buscar las imágenes de referencia de la lista 1 para otro bloque de referencia para la PU. La unidad de estimación de movimiento 122 puede entonces generar índices de referencia que indican las imágenes de referencia en la lista 0 y la lista 1 que contienen los bloques de referencia y los vectores de movimiento que indican desplazamientos espaciales entre los bloques de referencia y la PU. La unidad de estimación de movimiento 122 puede emitir los índices de referencia y los vectores de movimiento de la PU como la información de movimiento de la PU. La unidad de compensación de movimiento 124 puede generar el bloque de vídeo predicho de la PU basándose en los bloques de referencia indicados por la información de movimiento de la PU.

[0080] En algunos casos, la unidad de estimación de movimiento 122 no emite un conjunto completo de información de movimiento para una PU a la unidad de codificación por entropía 116. En lugar de eso, la unidad de estimación de movimiento 122 puede señalar la información de movimiento de una PU con referencia a la información de movimiento de otra PU. Por ejemplo, la unidad de estimación de movimiento 122 puede determinar que la información de movimiento de la PU es suficientemente similar a la información de movimiento de una PU próxima. En este ejemplo, la unidad de estimación de movimiento 122 puede indicar, en una estructura sintáctica asociada con la PU, un valor que indica al descodificador de vídeo 30 que la PU tiene la misma información de movimiento que la PU próxima. En otro ejemplo, la unidad de estimación de movimiento 122 puede identificar, en una estructura sintáctica asociada a la PU, una PU próxima y una diferencia de vectores de movimiento (MVD). La diferencia de vectores de movimiento indica una diferencia entre el vector de movimiento de la PU y el vector de movimiento de la PU próxima indicada. El descodificador de vídeo 30 puede usar el vector de movimiento de la PU próxima indicada y la diferencia de vectores de movimiento para determinar el vector de movimiento de la PU. Haciendo referencia a la información de movimiento de una primera PU cuando se señala la información de movimiento de una segunda PU, el codificador de vídeo 20 puede ser capaz de señalar la información de movimiento de la segunda PU usando menos bits.

[0081] Como se analizará más adelante a continuación con referencia a la FIG. 5, la unidad de procesamiento de predicción 100 puede configurarse para codificar (por ejemplo, codificar o descodificar) la PU (o cualquier otra capa de referencia y/o bloques de capa de mejora o unidades de vídeo) realizando los procedimientos ilustrados en la FIG. 5. Por ejemplo, la unidad de interpredicción 121 (por ejemplo, a través de la unidad de estimación de movimiento 122 y/o la unidad de compensación de movimiento 124), la unidad de intrapredicción 126, o la unidad de predicción entre capas 128 pueden configurarse para realizar los procedimientos ilustrados en la FIG. 5 ya sea juntos o por separado.

[0082] Como parte de realizar una operación de codificación en una CU, la unidad de intrapredicción 126 puede llevar a cabo la intrapredicción en las PU de la CU. La intrapredicción puede proporcionar compresión espacial. Cuando la unidad de intrapredicción 126 realiza la intrapredicción en una PU, la unidad de intrapredicción 126 puede

generar datos de predicción para la PU basados en muestras descodificadas de otras PU en la misma imagen. Los datos de predicción para la PU pueden incluir un bloque de vídeo predicho y varios elementos sintácticos. La unidad de intrapredicción 126 puede realizar la intrapredicción en PU en fragmentos I, fragmentos P y fragmentos B.

5 **[0083]** Para realizar la intrapredicción en una PU, la unidad de intrapredicción 126 puede usar múltiples modos de intrapredicción para generar múltiples conjuntos de datos de predicción para la PU. Cuando la unidad de intrapredicción 126 utiliza un modo de intrapredicción para generar un conjunto de datos de predicción para la PU, la unidad de intrapredicción 126 puede extender muestras de bloques de vídeo de PU próximas a través del bloque de vídeo de la PU en una dirección y/o gradiente asociados con el modo de intrapredicción. Las PU próximas pueden estar arriba, arriba y a la derecha, arriba y a la izquierda o a la izquierda de la PU, suponiendo un orden de codificación de izquierda a derecha, de arriba a abajo para las PU, CU y bloques de árbol. La unidad de intrapredicción 126 puede usar varios números de modos de intrapredicción, por ejemplo, 33 modos de intrapredicción direccional, dependiendo del tamaño de la PU.

15 **[0084]** La unidad de procesamiento de predicción 100 puede seleccionar los datos de predicción para una PU entre los datos de predicción generados por la unidad de compensación de movimiento 124 para la PU o los datos de predicción generados por la unidad de intrapredicción 126 para la PU. En algunos ejemplos, la unidad de procesamiento de predicción 100 selecciona los datos de predicción para la PU basándose en las métricas de velocidad/distorsión de los conjuntos de datos de predicción.

20 **[0085]** Si la unidad de procesamiento de predicción 100 selecciona datos de predicción generados por la unidad de intrapredicción 126, la unidad de procesamiento de predicción 100 puede señalar el modo de intrapredicción que se utilizó para generar los datos de predicción para las PU, es decir, el modo de intrapredicción seleccionado. La unidad de procesamiento de predicción 100 puede señalar el modo de intrapredicción seleccionado de varias maneras. Por ejemplo, es probable que el modo de intrapredicción seleccionado sea el mismo que el modo de intrapredicción de una PU próxima. En otras palabras, el modo de intrapredicción de la PU próxima puede ser el modo más probable para la PU actual. De este modo, la unidad de procesamiento de predicción 100 puede generar un elemento sintáctico para indicar que el modo de intrapredicción seleccionado es el mismo que el modo de intrapredicción de la PU próxima.

30 **[0086]** Como se analizó anteriormente, el codificador de vídeo 20 puede incluir la unidad de predicción entre capas 128. La unidad de predicción entre capas 128 está configurada para predecir un bloque actual (por ejemplo, un bloque actual en la EL) usando una o más capas diferentes que están disponibles en SVC (por ejemplo, una capa base o de referencia). Dicha predicción se puede denominar predicción entre capas. La unidad de predicción entre capas 128 utiliza procedimientos de predicción para reducir la redundancia entre capas, mejorando así la eficacia de codificación y reduciendo los requisitos de recursos informáticos. Algunos ejemplos de predicción entre capas incluyen la predicción intra entre capas, la predicción de movimiento entre capas y la predicción residual entre capas. La predicción intra entre capas utiliza la reconstrucción de bloques ubicados conjuntamente en la capa base para predecir el bloque actual en la capa de mejora. La predicción de movimiento entre capas usa la información de movimiento de la capa base para predecir el movimiento en la capa de mejora. La predicción residual entre capas usa el residuo de la capa base para predecir el residuo de la capa de mejora. Cada uno de los esquemas de predicción entre capas se analiza a continuación con mayor detalle.

45 **[0087]** Después de que la unidad de procesamiento de predicción 100 seleccione los datos de predicción para las PU de una CU, la unidad de generación residual 102 puede generar datos residuales para la CU restando (por ejemplo, indicado por el signo menos) los bloques de vídeo predichos de las PU de la CU del bloque de vídeo de la CU. Los datos residuales de una CU pueden incluir bloques de vídeo residuales 2D que corresponden a diferentes componentes de muestra de las muestras en el bloque de vídeo de la CU. Por ejemplo, los datos residuales pueden incluir un bloque de vídeo residual que corresponde a diferencias entre componentes de luminancia de muestras en los bloques de vídeo predichos de las PU de la CU y componentes de luminancia de muestras en el bloque de vídeo original de la CU. Además, los datos residuales de la CU pueden incluir bloques de vídeo residuales que corresponden a las diferencias entre componentes de crominancia de muestras en los bloques de vídeo predichos de las PU de la CU y los componentes de crominancia de las muestras en el bloque de vídeo original de la CU.

55 **[0088]** La unidad de procesamiento de predicción 100 puede realizar la división en árbol cuaternario para dividir los bloques de vídeo residuales de una CU en subbloques. Cada bloque de vídeo residual no dividido puede estar asociado a una TU diferente de la CU. Los tamaños y posiciones de los bloques de vídeo residuales asociados a las TU de una CU pueden basarse, o no, en los tamaños y posiciones de bloques de vídeo asociados a las PU de la CU. Una estructura en árbol cuaternario conocida como "árbol cuaternario residual" (RQT) puede incluir nodos asociados a cada uno de los bloques de vídeo residuales. Las TU de una CU pueden corresponder a nodos hoja del RQT.

60 **[0089]** La unidad de procesamiento de transformación 104 puede generar uno o más bloques de coeficientes de transformación para cada TU de una CU aplicando una o más transformaciones a un bloque de vídeo residual asociado con la TU. Cada uno de los bloques de coeficientes de transformación puede ser una matriz 2D de coeficientes de transformación. La unidad de procesamiento de transformación 104 puede aplicar varias transformaciones al bloque de vídeo residual asociado con una TU. Por ejemplo, la unidad de procesamiento de

65

transformación 104 puede aplicar una transformación de coseno discreta (DCT), una transformación direccional o una transformación conceptualmente similar al bloque de vídeo residual asociado con una TU.

5 [0090] Después de que la unidad de procesamiento de transformación 104 genere un bloque de coeficientes de transformación asociado con una TU, la unidad de cuantificación 106 puede cuantificar los coeficientes de transformación en el bloque de coeficientes de transformación. La unidad de cuantificación 106 puede cuantificar un bloque de coeficientes de transformación asociado con una TU de una CU basándose en un valor de QP asociado con la CU.

10 [0091] El codificador de vídeo 20 puede asociar un valor QP con una CU de varias maneras. Por ejemplo, el codificador de vídeo 20 puede realizar un análisis de velocidad-distorsión en un bloque de árbol asociado con la CU. En el análisis de velocidad-distorsión, el codificador de vídeo 20 puede generar múltiples representaciones codificadas del bloque de árbol realizando una operación de codificación varias veces en el bloque de árbol. El codificador de vídeo 20 puede asociar diferentes valores de QP con la CU cuando el codificador de vídeo 20 genera diferentes representaciones codificadas del bloque de árbol. El codificador de vídeo 20 puede señalar que un valor QP determinado está asociado con la CU cuando el valor QP determinado está asociado con la CU en una representación codificada del bloque de árbol que tiene una velocidad de bits y una métrica de distorsión más bajas.

20 [0092] La unidad de cuantificación inversa 108 y la unidad de transformación inversa 110 pueden aplicar la cuantificación inversa y las transformaciones inversas al bloque de coeficientes de transformación, respectivamente, para reconstruir un bloque de vídeo residual a partir del bloque de coeficientes de transformación. La unidad de reconstrucción 112 puede añadir el bloque de vídeo residual reconstruido a las muestras correspondientes de uno o más bloques de vídeo predichos generados por la unidad de procesamiento de predicción 100 para producir un bloque de vídeo reconstruido asociado con una TU. Mediante la reconstrucción de bloques de vídeo para cada TU de una CU de esta manera, el codificador de vídeo 20 puede reconstruir el bloque de vídeo de la CU.

30 [0093] Después de que la unidad de reconstrucción 112 reconstruya el bloque de vídeo de una CU, la unidad de filtro 113 puede realizar una operación de desbloqueo para reducir los artefactos de bloqueo en el bloque de vídeo asociado con la CU. Después de realizar las una o más operaciones de desbloqueo, la unidad de filtro 113 puede almacenar el bloque de vídeo reconstruido de la CU en la memoria intermedia de imágenes descodificadas 114. La unidad de estimación de movimiento 122 y la unidad de compensación de movimiento 124 pueden usar una imagen de referencia que contiene el bloque de vídeo reconstruido para realizar la interpredicción en las PU de las imágenes posteriores. Además, la unidad de intrapredicción 126 puede usar bloques de vídeo reconstruidos en la memoria intermedia de imágenes descodificadas 114 para realizar la intrapredicción en otras PU en la misma imagen que la CU.

40 [0094] La unidad de codificación por entropía 116 puede recibir datos de otros componentes funcionales del codificador de vídeo 20. Por ejemplo, la unidad de codificación por entropía 116 puede recibir bloques de coeficientes de la unidad de cuantificación 106 y puede recibir elementos sintácticos de la unidad de procesamiento de predicción 100. Cuando la unidad de codificación por entropía 116 recibe los datos, la unidad de codificación por entropía 116 puede realizar una o más operaciones de codificación por entropía para generar datos codificados por entropía. Por ejemplo, el codificador de vídeo 20 puede realizar una operación de codificación de longitud variable adaptable al contexto (CAVLC), una operación CABAC, una operación de codificación de longitud variable a variable (V2V), una operación de codificación aritmética binaria adaptable al contexto basada en sintaxis (SBAC), una operación de codificación por entropía de división de intervalo de probabilidad (PIPE) u otro tipo de operación de codificación por entropía en los datos. La unidad de codificación por entropía 116 puede emitir un flujo de bits que incluye los datos codificados por entropía.

50 [0095] Como parte de realizar una operación de codificación por entropía en los datos, la unidad de codificación por entropía 116 puede seleccionar un modelo de contexto. Si la unidad de codificación por entropía 116 está realizando una operación CABAC, el modelo de contexto puede indicar las estimaciones de las probabilidades de que un bin particular tenga unos valores particulares. En el contexto de CABAC, el término "bin" se utiliza para hacer referencia a un bit de una versión binarizada de un elemento sintáctico.

55 **Codificador de vídeo de múltiples capas**

60 [0096] La **FIG. 2B** es un diagrama de bloques que ilustra un ejemplo de un codificador de vídeo de múltiples capas 23 (también denominado simplemente codificador de vídeo 23) que puede implementar técnicas de acuerdo con los aspectos descritos en esta divulgación. El codificador de vídeo 23 puede configurarse para procesar tramas de vídeo de múltiples capas, como para codificación SHVC y de múltiples visualizaciones. Además, el codificador de vídeo 23 puede configurarse para realizar cualquiera o todas las técnicas de esta divulgación.

65 [0097] El codificador de vídeo 23 incluye un codificador de vídeo 20A y un codificador de vídeo 20B, cada uno de los cuales puede configurarse como el codificador de vídeo 20 y puede realizar las funciones descritas anteriormente con respecto al codificador de vídeo 20. Además, como se indica por la reutilización de los números de referencia, los codificadores de vídeo 20A y 20B pueden incluir, al menos, algunos de los sistemas y subsistemas como el codificador

de vídeo 20. Aunque el codificador de vídeo 23 se ilustra como que incluye dos codificadores de vídeo 20A y 20B, el codificador de vídeo 23 no está limitado como tal y puede incluir cualquier número de capas de codificador de vídeo 20. En algunos modos de realización, el codificador de vídeo 23 puede incluir un codificador de vídeo 20 para cada imagen o trama en una unidad de acceso. Por ejemplo, una unidad de acceso que incluye cinco imágenes puede ser procesada o codificada por un codificador de vídeo que incluye cinco capas de codificador. En algunos modos de realización, el codificador de vídeo 23 puede incluir más capas de codificador que tramas en una unidad de acceso. En algunos de dichos casos, algunas de las capas del codificador de vídeo pueden estar inactivas al procesar algunas unidades de acceso.

[0098] Además de los codificadores de vídeo 20A y 20B, el codificador de vídeo 23 puede incluir una unidad de remuestreo 90. La unidad de remuestreo 90 puede, en algunos casos, realizar muestreo ascendente de una capa base de una trama de vídeo recibida para, por ejemplo, crear una capa de mejora. La unidad de remuestreo 90 puede realizar muestreo ascendente de la información particular asociada con la capa base recibida de una trama, pero no otra información. Por ejemplo, la unidad de remuestreo 90 puede realizar muestreo ascendente del tamaño espacial o el número de píxeles de la capa base, pero el número de fragmentos o el recuento de orden de imágenes puede permanecer constante. En algunos casos, la unidad de remuestreo 90 puede no procesar el vídeo recibido y/o puede ser opcional. Por ejemplo, en algunos casos, la unidad de procesamiento de predicción 100 puede realizar una interpolación. En algunos modos de realización, la unidad de remuestreo 90 está configurada para realizar muestreo ascendente de una capa y reorganizar, redefinir, modificar o ajustar uno o más fragmentos para cumplir con un conjunto de reglas de límite de fragmento y/o reglas de escaneo de trama. Aunque se describe principalmente como un muestreo ascendente de una capa base o de una capa inferior en una unidad de acceso, en algunos casos, la unidad de remuestreo 90 puede realizar muestreo descendente de una capa. Por ejemplo, si durante la transmisión de un vídeo en tiempo real se reduce el ancho de banda, una trama puede sufrir muestreo descendente en lugar de muestreo ascendente.

[0099] La unidad de remuestreo 90 puede configurarse para recibir una imagen o trama (o información de imagen asociada con la imagen) desde la memoria intermedia de imágenes descodificadas 114 del codificador de capa inferior (por ejemplo, el codificador de vídeo 20A) y para realizar muestreo ascendente de la imagen (o la información de imagen recibida). Esta imagen interpolada puede entonces proporcionarse a la unidad de procesamiento de predicción 100 de un codificador de capa superior (por ejemplo, el codificador de vídeo 20B) configurado para codificar una imagen en la misma unidad de acceso que el codificador de capa inferior. En algunos casos, el codificador de capa superior es una capa eliminada del codificador de capa inferior. En otros casos, puede haber uno o más codificadores de capa superior entre el codificador de vídeo de capa 0 y el codificador de capa 1 de la **FIG. 2B**.

[0100] En algunos casos, la unidad de remuestreo 90 se puede omitir o anular. En dichos casos, la imagen de la memoria intermedia de imágenes descodificadas 114 del codificador de vídeo 20A se puede proporcionar directamente, o al menos sin proporcionarse a la unidad de remuestreo 90, a la unidad de procesamiento de predicción 100 del codificador de vídeo 20B. Por ejemplo, si los datos de vídeo proporcionados al codificador de vídeo 20B y la imagen de referencia de la memoria intermedia de imágenes descodificadas 114 del codificador de vídeo 20A son del mismo tamaño o resolución, la imagen de referencia puede proporcionarse al codificador de vídeo 20B sin ningún remuestreo.

[0101] En algunos modos de realización, el codificador de vídeo 23 realiza muestreo descendente de los datos de vídeo a proporcionar al codificador de capa inferior usando la unidad de muestreo descendente 94 antes de proporcionar los datos de vídeo al codificador de vídeo 20A. De forma alternativa, la unidad de muestreo descendente 94 puede ser una unidad de remuestreo 90 capaz de realizar un muestreo ascendente o muestreo descendente de los datos de vídeo. En otros modos de realización más, la unidad de muestreo descendente 94 puede omitirse.

[0102] Como se ilustra en la **FIG. 2B**, el codificador de vídeo 23 puede incluir, además, un multiplexor 98, o mux. El mux 98 puede enviar un flujo de bits combinado desde el codificador de vídeo 23. El flujo de bits combinado se puede crear tomando un flujo de bits de cada uno de los codificadores de vídeo 20A y 20B y alternando qué flujo de bits se envía en un momento determinado. Mientras que en algunos casos los bits de los dos (o más en el caso de más de dos capas de codificador de vídeo) flujos de bits pueden alternarse un bit a la vez, en muchos casos los flujos de bits se combinan de manera diferente. Por ejemplo, el flujo de bits de salida puede crearse alternando el flujo de bits seleccionado un bloque a la vez. En otro ejemplo, el flujo de bits de salida se puede crear enviando una proporción de bloques diferente a 1:1 de cada uno de los codificadores de vídeo 20A y 20B. Por ejemplo, pueden enviarse dos bloques desde el codificador de vídeo 20B para cada salida de bloque del codificador de vídeo 20A. En algunos modos de realización, el flujo de salida del mux 98 puede preprogramarse. En otros modos de realización, el mux 98 puede combinar los flujos de bits de los codificadores de vídeo 20A, 20B en base a una señal de control recibida desde un sistema externo al codificador de vídeo 23, tal como desde un procesador en un dispositivo de origen que incluye el dispositivo de origen 12. La señal de control puede generarse en base a la resolución o velocidad de bits de un vídeo desde la fuente de vídeo 18, en base al ancho de banda del enlace 16, en base a una suscripción asociada a un usuario (por ejemplo, una suscripción de pago frente a una suscripción gratuita) o en base a cualquier otro factor para determinar una salida de resolución deseada del codificador de vídeo 23.

Descodificador de vídeo

5 [0103] La FIG. 3A es un diagrama de bloques que ilustra un ejemplo del decodificador de vídeo 30 que puede implementar las técnicas de acuerdo a aspectos descritos en esta divulgación. El decodificador de vídeo 30 puede configurarse para procesar una sola capa de una trama de vídeo, tal como para HEVC. Además, el decodificador de vídeo 30 puede configurarse para realizar cualquiera o todas las técnicas de esta divulgación, que incluyen, pero no se limitan a, los procedimientos para inferir NoOutputOfPriorPicsFlag y procesos relacionados descritos con mayor detalle anteriormente y posteriormente con respecto a las FIGS. 4 y 5. Como ejemplo, la unidad de compensación de movimiento 162 y/o la unidad de intrapredicción 164 pueden configurarse para realizar cualquiera o todas las técnicas descritas en esta divulgación. En un modo de realización, el decodificador de vídeo 30 puede incluir opcionalmente una unidad de predicción 166 entre capas que está configurada para realizar cualquiera o todas las técnicas descritas en esta divulgación. En otros modos de realización, la predicción entre capas puede realizarse mediante la unidad de procesamiento de predicción 152 (por ejemplo, la unidad de compensación de movimiento 162 y/o la unidad de intrapredicción 164), en cuyo caso la unidad de predicción entre capas 166 puede omitirse. Sin embargo, los aspectos de esta divulgación no se limitan a lo anterior. En algunos ejemplos, las técnicas descritas en esta divulgación pueden compartirse entre los diversos componentes del decodificador de vídeo 30. En algunos ejemplos, adicionalmente o de forma alternativa, un procesador (no mostrado) puede configurarse para realizar cualquiera o todas las técnicas descritas en esta divulgación.

20 [0104] Con fines de explicación, esta divulgación describe el decodificador de vídeo 30 en el contexto de la codificación HEVC. Sin embargo, las técnicas de esta divulgación pueden ser aplicables a otras normas o procedimientos de codificación. El ejemplo representado en la FIG. 3A es para un códec de una sola capa. Sin embargo, como se describirá adicionalmente con respecto a la FIG. 3B, parte o la totalidad del decodificador de vídeo 30 puede duplicarse para procesar un códec multicapa.

25 [0105] En el ejemplo de la FIG. 3A, el decodificador de vídeo 30 incluye una pluralidad de componentes funcionales. Los componentes funcionales del decodificador de vídeo 30 incluyen una unidad de descodificación por entropía 150, una unidad de procesamiento de predicción 152, una unidad de cuantificación inversa 154, una unidad de transformada inversa 156, una unidad de reconstrucción 158, una unidad de filtro 159 y una memoria intermedia de imágenes descodificadas 160. La unidad de procesamiento de predicción 152 incluye una unidad de compensación de movimiento 162, una unidad de intrapredicción 164 y una unidad de predicción entre capas 166. En algunos ejemplos, el decodificador de vídeo 30 puede realizar una pasada de descodificación, en general, recíproca a la pasada de codificación descrita con respecto al codificador de vídeo 20 de la FIG. 2A. En otros ejemplos, el decodificador de vídeo 30 puede incluir más, menos o diferentes componentes funcionales.

35 [0106] El decodificador de vídeo 30 puede recibir un flujo de bits que comprende datos de vídeo codificados. El flujo de bits puede incluir una pluralidad de elementos sintácticos. Cuando el decodificador de vídeo 30 recibe el flujo de bits, la unidad de descodificación por entropía 150 puede realizar una operación de análisis sintáctico en el flujo de bits. Como resultado de realizar la operación de análisis sintáctico en el flujo de bits, la unidad de descodificación por entropía 150 puede extraer elementos sintácticos del flujo de bits. Como parte de realizar la operación de análisis sintáctico, la unidad de descodificación por entropía 150 puede realizar la descodificación por entropía de elementos sintácticos sometidos a codificación por entropía en el flujo de bits. La unidad de procesamiento de predicción 152, la unidad de cuantificación inversa 154, la unidad de transformación inversa 156, la unidad de reconstrucción 158 y la unidad de filtro 159 pueden realizar una operación de reconstrucción que genera datos de vídeo descodificados basados en los elementos sintácticos extraídos del flujo de bits.

45 [0107] Como se analizó anteriormente, el flujo de bits puede comprender una serie de unidades NAL. Las unidades NAL del flujo de bits pueden incluir unidades NAL de conjunto de parámetros de vídeo, unidades NAL de conjunto de parámetros de secuencia, unidades NAL de conjunto de parámetros de imagen, unidades NAL SEI, etc. Como parte de realizar la operación de análisis sintáctico en el flujo de bits, la unidad de descodificación por entropía 150 puede realizar operaciones de análisis sintáctico que extraen y descodifican por entropía conjuntos de parámetros de secuencia a partir de unidades NAL de conjuntos de parámetros de secuencia, conjuntos de parámetros de imagen a partir de unidades NAL de conjuntos de parámetros de imagen, datos SEI a partir de unidades NAL SEI, etc.

55 [0108] Además, las unidades NAL del flujo de bits pueden incluir unidades NAL de fragmentos codificados. Como parte de realizar la operación de análisis sintáctico en el flujo de bits, la unidad de descodificación por entropía 150 puede realizar operaciones de análisis sintáctico que extraen y descodifican por entropía fragmentos codificados a partir de las unidades de NAL de fragmentos codificados. Cada uno de los fragmentos codificados puede incluir una cabecera de fragmento y datos de fragmento. La cabecera de fragmento puede contener elementos sintácticos pertenecientes a un fragmento. Los elementos sintácticos de la cabecera de fragmento pueden incluir un elemento sintáctico que identifica un conjunto de parámetros de imagen asociado a una imagen que contiene el fragmento. La unidad de descodificación por entropía 150 puede realizar operaciones de descodificación por entropía, tales como operaciones de descodificación de CABAC, en los elementos sintácticos de la cabecera del fragmento codificado para recuperar la cabecera del fragmento.

65 [0109] Como parte de extraer los datos del fragmento de unidades NAL de fragmentos codificados, la unidad de descodificación por entropía 150 puede realizar operaciones de análisis sintáctico que extraen los elementos

sintácticos de las CU codificadas en los datos del fragmento. Los elementos sintácticos extraídos pueden incluir elementos sintácticos asociados a bloques de coeficientes de transformación. La unidad de descodificación por entropía 150 puede entonces realizar operaciones de descodificación de CABAC en algunos de los elementos sintácticos.

5 **[0110]** Después de que la unidad de descodificación por entropía 150 realice una operación de análisis sintáctico en una CU no dividida, el descodificador de vídeo 30 puede realizar una operación de reconstrucción en la CU no dividida. Para realizar la operación de reconstrucción en una CU no dividida, el descodificador de vídeo 30 puede realizar una operación de reconstrucción en cada TU de la CU. Realizando la operación de reconstrucción para cada
10 TU de la CU, el descodificador de vídeo 30 puede reconstruir un bloque de vídeo residual asociado con la CU.

[0111] Como parte de realizar una operación de reconstrucción en una TU, la unidad de cuantificación inversa 154 puede cuantificar de forma inversa, es decir, descuantificar, un bloque de coeficientes de transformación asociado a la TU. La unidad de cuantificación inversa 154 puede cuantificar de forma inversa el bloque de coeficientes de transformación de una manera similar a los procesos de cuantificación inversa propuestos para HEVC o definidos por la norma de descodificación H.264. La unidad de cuantificación inversa 154 puede usar un parámetro de cuantificación QP calculado por el codificador de vídeo 20 para una CU del bloque de coeficientes de transformación para determinar un grado de cuantificación y, del mismo modo, un grado de cuantificación inversa para la unidad de cuantificación inversa 154 a aplicar.
15
20

[0112] Después de que la unidad de cuantificación inversa 154 cuantifique de forma inversa un bloque de coeficientes de transformación, la unidad de transformación inversa 156 puede generar un bloque de vídeo residual para la TU asociada con el bloque de coeficientes de transformación. La unidad de transformación inversa 156 puede aplicar una transformación inversa al bloque de coeficientes de transformación para generar el bloque de vídeo residual para la TU. Por ejemplo, la unidad de transformación inversa 156 puede aplicar una DCT inversa, una transformación de número entero inversa, una transformación de Karhunen-Loeve (KLT) inversa, una transformación de rotación inversa, una transformación direccional inversa u otra transformación inversa al bloque de coeficientes de transformación. En algunos ejemplos, la unidad de transformación inversa 156 puede determinar una transformación inversa para aplicar al bloque de coeficientes de transformación basándose en la señalización del codificador de vídeo 20. En dichos ejemplos, la unidad de transformación inversa 156 puede determinar la transformación inversa basándose en una transformación señalizada en el nodo raíz de un árbol cuaternario para un bloque de árbol asociado al bloque de coeficientes de transformación. En otros ejemplos, la unidad de transformación inversa 156 puede inferir la transformación inversa a partir de una o más características de codificación, tales como tamaño de bloque, modo de codificación o similares. En algunos ejemplos, la unidad de transformación inversa 156 puede aplicar una transformación inversa en cascada.
25
30
35

[0113] En algunos ejemplos, la unidad de compensación de movimiento 162 puede refinar el bloque de vídeo predicho de una PU mediante interpolación basándose en filtros de interpolación. Los identificadores para los filtros de interpolación que van a usarse para la compensación de movimiento con una precisión de submuestra pueden incluirse en los elementos sintácticos. La unidad de compensación de movimiento 162 puede usar los mismos filtros de interpolación usados por el codificador de vídeo 20 durante la generación del bloque de vídeo predicho de la PU para calcular valores interpolados para muestras de subenteros de un bloque de referencia. La unidad de compensación de movimiento 162 puede determinar los filtros de interpolación usados por el codificador de vídeo 20 de acuerdo con la información sintáctica recibida y usar los filtros de interpolación para producir el bloque de vídeo predicho.
40
45

[0114] Como se analiza adicionalmente más adelante con referencia a la **FIG. 5**, la unidad de procesamiento de predicción 152 puede codificar (por ejemplo, codificar o descodificar) la PU (o cualquier otra capa de referencia y/o bloques de capa de mejora o unidades de vídeo) realizando los procedimientos ilustrados en la **FIG. 5**. Por ejemplo, la unidad de compensación de movimiento 162, la unidad de intrapredicción 164 o la unidad de predicción entre capas 166 pueden configurarse para realizar los procedimientos ilustrados en la **FIG. 5**, ya sea juntos o por separado.
50

[0115] Si se codifica una PU usando intrapredicción, la unidad de intrapredicción 164 puede realizar la intrapredicción para generar un bloque de vídeo predicho para la PU. Por ejemplo, la unidad de intrapredicción 164 puede determinar un modo de intrapredicción para la PU basándose en los elementos sintácticos del flujo de bits. El flujo de bits puede incluir elementos sintácticos que la unidad de intrapredicción 164 puede usar para determinar el modo de intrapredicción de la PU.
55

[0116] En algunos casos, los elementos sintácticos pueden indicar que la unidad de intrapredicción 164 tiene que usar el modo de intrapredicción de otra PU para determinar el modo de intrapredicción de la PU actual. Por ejemplo, puede ser probable que el modo de intrapredicción de la PU actual sea el mismo que el modo de intrapredicción de una PU próxima. En otras palabras, el modo de intrapredicción de la PU próxima puede ser el modo más probable para la PU actual. Por lo tanto, en este ejemplo, el flujo de bits puede incluir un pequeño elemento sintáctico que indica que el modo de intrapredicción de la PU es el mismo que el modo de intrapredicción de la PU próxima. La unidad de intrapredicción 164 puede entonces usar el modo de intrapredicción para generar datos de predicción (por ejemplo, muestras predichas) para la PU basándose en los bloques de vídeo de las PU espacialmente próximas.
60
65

[0117] Como se analizó anteriormente, el descodificador de vídeo 30 también puede incluir la unidad de predicción entre capas 166. La unidad de predicción entre capas 166 está configurada para predecir un bloque actual (por ejemplo, un bloque actual en la EL) usando una o más capas diferentes que están disponibles en SVC (por ejemplo, una capa base o de referencia). Dicha predicción se puede denominar predicción entre capas. La unidad de predicción entre capas 166 utiliza procedimientos de predicción para reducir la redundancia entre capas, mejorando así la eficacia de codificación y reduciendo los requisitos de recursos informáticos. Algunos ejemplos de predicción entre capas incluyen la predicción intra entre capas, la predicción de movimiento entre capas y la predicción residual entre capas. La predicción intra entre capas utiliza la reconstrucción de bloques ubicados conjuntamente en la capa base para predecir el bloque actual en la capa de mejora. La predicción de movimiento entre capas usa la información de movimiento de la capa base para predecir el movimiento en la capa de mejora. La predicción residual entre capas usa el residuo de la capa base para predecir el residuo de la capa de mejora. Cada uno de los esquemas de predicción entre capas se analiza a continuación con mayor detalle.

[0118] La unidad de reconstrucción 158 puede usar los bloques de vídeo residuales asociados a las TU de una CU y los bloques de vídeo predichos de las PU de la CU, es decir, datos de intrapredicción o datos de interpredicción, según corresponda, para reconstruir el bloque de vídeo de la CU. De este modo, el descodificador de vídeo 30 puede generar un bloque de vídeo predicho y un bloque de vídeo residual basándose en los elementos sintácticos del flujo de bits y puede generar un bloque de vídeo basándose en el bloque de vídeo predicho y el bloque de vídeo residual.

[0119] Después de que la unidad de reconstrucción 158 reconstruya el bloque de vídeo de la CU, la unidad de filtro 159 puede realizar una operación de desbloqueo para reducir los artefactos de bloqueo asociados a la CU. Después de que la unidad de filtro 159 realiza una operación de desbloqueo para reducir los artefactos de bloqueo asociados con la CU, el descodificador de vídeo 30 puede almacenar el bloque de vídeo de la CU en la memoria intermedia de imágenes descodificadas 160. La memoria intermedia de imágenes descodificadas 160 puede proporcionar imágenes de referencia para una posterior compensación de movimiento, intrapredicción y presentación en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la FIG. 1A o 1B. Por ejemplo, el descodificador de vídeo 30 puede realizar, basándose en los bloques de vídeo de la memoria intermedia de imágenes descodificadas 160, operaciones de intrapredicción o de interpredicción en las PU de otras CU.

Descodificador de múltiples capas

[0120] La FIG. 3B es un diagrama de bloques que ilustra un ejemplo de descodificador 33 de vídeo de múltiples capas (también denominado simplemente descodificador de vídeo 33) que puede implementar técnicas de acuerdo con los aspectos descritos en esta divulgación. El descodificador de vídeo 33 puede configurarse para procesar tramas de vídeo de múltiples capas, como para codificación SHVC y de múltiples visualizaciones. Además, el descodificador de vídeo 33 se puede configurar para realizar cualquiera o todas las técnicas de esta divulgación.

[0121] El descodificador de vídeo 33 incluye un descodificador de vídeo 30A y un descodificador de vídeo 30B, cada uno de los cuales puede configurarse como el descodificador de vídeo 30 y puede realizar las funciones descritas anteriormente con respecto al descodificador de vídeo 30. Además, como se indica por la reutilización de los números de referencia, los descodificadores de vídeo 30A y 30B pueden incluir al menos algunos de los sistemas y subsistemas como el descodificador de vídeo 30. Aunque el descodificador de vídeo 33 se ilustra como que incluye dos descodificadores de vídeo 30A y 30B, el descodificador de vídeo 33 no está limitado como tal y puede incluir cualquier número de capas de descodificador de vídeo 30. En algunos modos de realización, el descodificador de vídeo 33 puede incluir un descodificador de vídeo 30 para cada imagen o trama en una unidad de acceso. Por ejemplo, una unidad de acceso que incluye cinco imágenes puede procesarse o descodificarse mediante un descodificador de vídeo que incluye cinco capas de descodificador. En algunos modos de realización, el descodificador de vídeo 33 puede incluir más capas de descodificador que tramas en una unidad de acceso. En algunos de dichos casos, algunas de las capas del descodificador de vídeo pueden estar inactivas al procesar algunas unidades de acceso.

[0122] Además de los descodificadores de vídeo 30A y 30B, el descodificador de vídeo 33 puede incluir una unidad de muestreo ascendente 92. En algunos modos de realización, la unidad de muestreo ascendente 92 puede realizar muestreo ascendente de una capa base de una trama de vídeo recibida para crear una capa mejorada para ser añadida a la lista de imágenes de referencia para la trama o la unidad de acceso. Esta capa mejorada se puede almacenar en la memoria intermedia de imágenes descodificadas 160. En algunos modos de realización, la unidad de interpolación 92 puede incluir algunos o todos los modos de realización descritos con respecto a la unidad de remuestreo 90 de la FIG. 2A. En algunos modos de realización, la unidad de muestreo ascendente 92 está configurada para realizar muestreo ascendente de una capa y reorganizar, redefinir, modificar o ajustar uno o más fragmentos para cumplir con un conjunto de reglas de límite de fragmento y/o reglas de escaneo de trama. En algunos casos, la unidad de interpolación 92 puede ser una unidad de remuestreo configurada para interpolar y/o diezmar una capa de una trama de vídeo recibida.

[0123] La unidad de muestreo ascendente 92 puede estar configurada para recibir una imagen o trama (o información de imagen asociada a la imagen) de la memoria intermedia de imágenes descodificadas 160 del descodificador de capa inferior (por ejemplo, el descodificador de vídeo 30A) y para realizar muestreo ascendente de

la imagen (o la información de imagen recibida). Esta imagen interpolada puede entonces proporcionarse a la unidad de procesamiento de predicción 152 de un descodificador de capa superior (por ejemplo, el descodificador de vídeo 30B) configurado para descodificar una imagen en la misma unidad de acceso que el descodificador de capa inferior. En algunos casos, el descodificador de capa superior es una capa eliminada del descodificador de capa inferior. En otros casos, puede haber uno o más descodificadores de capa superior entre el descodificador de capa 0 y el descodificador de capa 1 de la **FIG. 3B**.

[0124] En algunos casos, la unidad de muestreo ascendente 92 se puede omitir o anular. En dichos casos, la imagen de la memoria intermedia de imágenes descodificadas 160 del descodificador de vídeo 30A se puede proporcionar directamente, o al menos sin proporcionarse a la unidad de interpolación 92, a la unidad de procesamiento de predicción 152 del descodificador de vídeo 30B. Por ejemplo, si los datos de vídeo proporcionados al descodificador de vídeo 30B y la imagen de referencia de la memoria intermedia de imágenes descodificadas 160 del descodificador de vídeo 30A son del mismo tamaño o resolución, la imagen de referencia puede proporcionarse al descodificador de vídeo 30B sin muestreo ascendente. Además, en algunos modos de realización, la unidad de muestreo ascendente 92 puede ser una unidad de remuestreo 90 configurada para realizar muestreo ascendente o muestreo descendente de la imagen de referencia recibida de la memoria intermedia de imágenes descodificadas 160 del descodificador de vídeo 30A.

[0125] Tal como se ilustra en la **FIG. 3B**, el descodificador de vídeo 33 puede incluir, además, un demultiplexor 99, o demux. El demux 99 puede dividir un flujo de bits de vídeo codificado en múltiples flujos de bits con cada flujo de bits enviado por el demux 99 que se proporciona a un descodificador de vídeo diferente 30A y 30B. Los múltiples flujos de bits pueden crearse al recibir un flujo de bits y cada uno de los descodificadores de vídeo 30A y 30B recibe una parte del flujo de bits en un momento determinado. Mientras que en algunos casos los bits del flujo de bits recibidos en el demux 99 pueden alternarse un bit a la vez entre cada uno de los descodificadores de vídeo (por ejemplo, los descodificadores de vídeo 30A y 30B en el ejemplo de la **FIG. 3B**), en muchos casos el flujo de bits se divide de manera diferente. Por ejemplo, el flujo de bits se puede dividir alternando qué descodificador de vídeo recibe el flujo de bits un bloque a la vez. En otro ejemplo, el flujo de bits puede dividirse por una proporción de bloques diferente a 1: 1 a cada uno de los descodificadores de vídeo 30A y 30B. Por ejemplo, pueden proporcionarse dos bloques al descodificador de vídeo 30B por cada bloque proporcionado al descodificador de vídeo 30A. En algunos modos de realización, la división del flujo de bits por el demux 99 puede preprogramarse. En otros modos de realización, el demux 99 puede dividir el flujo de bits en base a una señal de control recibida desde un sistema externo al descodificador de vídeo 33, tal como desde un procesador en un dispositivo de destino que incluye el módulo de destino 14. La señal de control puede generarse en base a la resolución o la velocidad de bits de un vídeo desde la interfaz de entrada 28, en base al ancho de banda del enlace 16, en base a una suscripción asociada a un usuario (por ejemplo, una suscripción de pago frente a una suscripción gratuita) o en base a cualquier otro factor para determinar una resolución obtenible por el descodificador de vídeo 33.

Imágenes de punto de acceso aleatorio interno (IRAP)

[0126] Algunos esquemas de codificación de vídeo pueden proporcionar diversos puntos de acceso aleatorios en todo el flujo de bits de manera que el flujo de bits puede descodificarse comenzando desde cualquiera de esos puntos de acceso aleatorios sin necesidad de descodificar ninguna imagen que preceda esos puntos de acceso aleatorios en el flujo de bits. En dichos esquemas de codificación de vídeo, todas las imágenes que siguen un punto de acceso aleatorio en el orden de salida (por ejemplo, incluyendo aquellas imágenes que están en la misma unidad de acceso que la imagen que proporciona el punto de acceso aleatorio) se pueden descodificar correctamente sin usar ninguna imagen que preceda al punto de acceso aleatorio. Por ejemplo, incluso si una parte del flujo de bits se pierde durante la transmisión o durante la descodificación, un descodificador puede reanudar la descodificación del flujo de bits comenzando desde el próximo punto de acceso aleatorio. La compatibilidad con el acceso aleatorio puede facilitar, por ejemplo, servicios dinámicos de transmisión, operaciones de búsqueda, cambio de canal, etc.

[0127] En algunos esquemas de codificación, dichos puntos de acceso aleatorio pueden proporcionarse mediante imágenes que se denominan imágenes de punto de acceso aleatorio interno (IRAP). Por ejemplo, un punto de acceso aleatorio (por ejemplo, proporcionado por una imagen IRAP de capa de mejora) en una capa de mejora ("capaA") que está contenida en una unidad de acceso ("auA") puede proporcionar acceso aleatorio específico de capa tal que para cada capa de referencia ("capaB") de la capaA (por ejemplo, una capa de referencia que es una capa que se usa para predecir la capaA) que tiene un punto de acceso aleatorio contenido en una unidad de acceso ("auB") que está en la capaB y precede a la auA en el orden de descodificación (o un punto de acceso aleatorio contenido en la auA), las imágenes en la capaA que siguen a la auB en el orden de salida (incluidas las imágenes ubicadas en la auB) se pueden descodificar correctamente sin necesidad de descodificar ninguna imagen en la capaA que preceda a la auB.

[0128] Las imágenes IRAP pueden codificarse usando la intrapredicción (por ejemplo, codificarse sin hacer referencia a otras imágenes), y pueden incluir, por ejemplo, imágenes instantáneas de actualización de descodificación (IDR), imágenes de acceso aleatorio limpio (CRA) e imágenes de acceso de enlace roto (BLA). Cuando hay una imagen IDR en el flujo de bits, todas las imágenes que preceden a la imagen IDR en el orden de descodificación no se utilizan para la predicción mediante las imágenes que siguen a la imagen IDR en el orden de descodificación. Cuando hay una imagen CRA en el flujo de bits, las imágenes que siguen a la imagen CRA pueden

o no usar imágenes que preceden a la imagen CRA en el orden de descodificación para la predicción. Dichas imágenes que siguen a la imagen CRA en el orden de descodificación, pero usan imágenes que preceden a la imagen CRA en el orden de descodificación, pueden denominarse imágenes principales omitidas de acceso aleatorio (RASL). Otro tipo de imagen que sigue a una imagen IRAP en el orden de descodificación y que precede a la imagen IRAP en el orden de salida es una imagen principal descodificable de acceso aleatorio (RADL), que puede no contener referencias a ninguna imagen que preceda a la imagen IRAP en el orden de descodificación. Las imágenes RASL pueden ser descartadas por el descodificador si las imágenes que preceden a la imagen CRA no están disponibles. Una imagen BLA indica al descodificador que las imágenes que preceden a la imagen BLA pueden no estar disponibles para el descodificador (por ejemplo, porque dos flujos de bits están unidos entre sí y la imagen BLA es la primera imagen del segundo flujo de bits en el orden de descodificación). Una unidad de acceso (por ejemplo, un grupo de imágenes que consisten en todas las imágenes codificadas asociadas con el mismo tiempo de salida en las múltiples capas) que contiene una imagen de capa base (por ejemplo, una imagen que tiene un valor de identificador (ID) de capa de 0) que es una imagen IRAP se puede denominar una unidad de acceso IRAP.

Alineación de capas cruzadas de imágenes IRAP

[0129] En SVC, es posible que no se requiera que las imágenes IRAP estén alineadas (por ejemplo, contenidas en la misma unidad de acceso) en diferentes capas. Por ejemplo, si las imágenes IRAP debían estar alineadas, cualquier unidad de acceso que contenga al menos una imagen IRAP solo contendría imágenes IRAP. Por otro lado, si no se requiriera que las imágenes IRAP estuvieran alineadas, en una sola unidad de acceso, una imagen (por ejemplo, en una primera capa) puede ser una imagen IRAP, y otra imagen (por ejemplo, en una segunda capa) puede ser una imagen no IRAP. Tener tales imágenes IRAP no alineadas en un flujo de bits puede proporcionar algunas ventajas. Por ejemplo, en un flujo de bits de dos capas, si hay más imágenes IRAP en la capa base que en la capa de mejora, en aplicaciones de radiodifusión y multidifusión, se puede lograr un bajo retardo de sintonización y una alta eficiencia de codificación.

[0130] En algunos esquemas de codificación de vídeo, se puede usar un recuento de orden de imágenes (POC) para realizar un seguimiento del orden relativo en el que se muestran las imágenes descodificadas. Algunos de estos esquemas de codificación pueden hacer que los valores de POC se restablezcan (por ejemplo, se establezcan en cero o se establezcan en algún valor señalado en el flujo de bits) siempre que ciertos tipos de imágenes aparezcan en el flujo de bits. Por ejemplo, los valores POC de ciertas imágenes IRAP pueden restablecerse, lo que hace que los valores POC de otras imágenes que preceden a esas imágenes IRAP en el orden de descodificación también se restablezcan. Esto puede ser problemático cuando no es necesario que las imágenes IRAP se alineen en las diferentes capas. Por ejemplo, cuando una imagen ("picA") es una imagen IRAP y otra imagen ("picB") en la misma unidad de acceso no es una imagen IRAP, el valor POC de una imagen ("picC"), que se restablece debido a que picA sea una imagen IRAP en la capa que contiene picA puede ser diferente del valor POC de una imagen ("picD") que no se restablece, en la capa que contiene picB, donde picC y picD están en la misma unidad de acceso. Esto hace que picC y picD tengan diferentes valores de POC aunque pertenezcan a la misma unidad de acceso (por ejemplo, el mismo tiempo de salida). Por lo tanto, en este ejemplo, el proceso de obtención para obtener los valores POC de picC y picD se puede modificar para producir valores POC que sean coherentes con la definición de valores POC y unidades de acceso.

Imagen de inicialización de capa (LIP)

[0131] En algunos esquemas de codificación, una imagen de inicialización de capa ("imagen LIP") se puede definir como una imagen que es una imagen IRAP que tiene un NoOutputOfPriorPicsFlag (por ejemplo, un indicador que indica que las imágenes RASL no se emitirán si se establece en 1 e indica que las imágenes RASL se emitirán si se establece en 0) se establece en 1 o una imagen que contenga una unidad de acceso IRAP inicial, que es una unidad de acceso IRAP en la que la imagen de la capa base (por ejemplo, una imagen que tiene una ID de capa de 0 o la ID de capa más pequeña definida en el flujo de bits) tiene el NoOutputOfPriorPicsFlag establecido en 1.

[0132] En algunos modos de realización, se puede activar un SPS en cada imagen LIP. Por ejemplo, cada imagen IRAP que tiene un NoOutputOfPriorPicsFlag establecido en 1 o cada imagen que está contenida en una unidad de acceso IRAP inicial, un nuevo SPS, que puede ser diferente (por ejemplo, especificar diferentes resoluciones de imagen, etc.) del SPS que fue activado previamente. Sin embargo, en un caso donde la imagen LIP no es una imagen IRAP (por ejemplo, cualquier imagen contenida en una unidad de acceso IRAP inicial) y la imagen de la capa base en la unidad de acceso IRAP inicial es una imagen IDR con un indicador NoCInrasOutputFlag (por ejemplo, un indicador que indica que las imágenes de omisión de acceso aleatorio de capa cruzada no se deben emitir si se establece en 1 e indica que las imágenes de omisión de acceso aleatorio de capa cruzada se deben emitir si se establece en 0) se establece en 0, la imagen LIP no debe permitirse activar un nuevo SPS. Si se activa un nuevo SPS en dicha imagen LIP en tal caso, particularmente cuando el contenido del SPS RBSP del nuevo SPS es diferente al del SPS que estaba activo antes de la unidad de acceso IRAP inicial, podría haber problemas en diferentes resoluciones de imagen y resistencia a errores. Por ejemplo, el nuevo SPS puede actualizar la resolución y usar la predicción temporal para referirse a imágenes de diferentes tamaños.

Desplazamiento y liberación de imágenes

[0133] Las imágenes que se decodifican (por ejemplo, para que puedan mostrarse o usarse para predecir otras imágenes) se almacenan en una memoria intermedia de imágenes decodificadas (DPB). Las imágenes que se enviarán pueden marcarse como "necesarias para la salida", y las imágenes que se utilizarán para predecir otras imágenes pueden marcarse como "utilizadas como referencia". Imágenes decodificadas que no están marcadas como "necesarias para la salida" ni como "utilizadas para referencia" (por ejemplo, imágenes que fueron marcadas inicialmente como "utilizadas para referencia" o "necesarias para la salida" pero posteriormente marcadas como "no utilizadas para referencia" o "no es necesario para la salida") puede estar presente en la DPB hasta que sean eliminados por el proceso de decodificación. En los decodificadores conformes con el orden de salida, el proceso de eliminación de imágenes de la DPB a menudo sigue inmediatamente la salida de imágenes que están marcadas como "necesarias para la salida". Este proceso de salida y posterior eliminación puede denominarse "desplazamiento".

[0134] También hay situaciones en las que el decodificador puede eliminar las imágenes en la DPB sin salida, a pesar de que estas imágenes pueden estar marcadas como "necesarias para la salida". Para facilitar la descripción en el presente documento, las imágenes decodificadas que están presentes en la DPB en el momento de decodificar una imagen IRAP (independientemente de si las imágenes decodificadas están marcadas como "necesarias para la salida" o "utilizadas para referencia") se denominan "imágenes DPB retardadas" asociadas con la imagen IRAP o "imágenes DPB retardadas asociadas" de la imagen IRAP. Algunos ejemplos de tales situaciones, en el contexto HEVC, se describen a continuación.

[0135] En un ejemplo, cuando una imagen CRA con NoRasOutputFlag igual a un valor de "1" está presente en el medio de un flujo de bits (por ejemplo, no la primera imagen en el flujo de bits), las imágenes DPB retrasadas asociadas con la imagen CRA no estarán salida y se eliminaría de la DPB. Es probable que tales situaciones ocurran en puntos de empalme, donde dos flujos de bits se unen y la primera imagen del último flujo de bits es una imagen CRA con NoRasOutputFlag igual a un valor de "1". En otro ejemplo, cuando una imagen IRAP picA que tiene NoRasOutputFlag igual a un valor de "1" y que no es una imagen CRA (por ejemplo, una imagen IDR) está presente en medio de un flujo de bits y la resolución de la imagen cambia en picA (por ejemplo, con la activación de un nuevo SPS), las imágenes DPB retrasadas asociadas de picA pueden eliminarse de la DPB antes de que puedan emitirse, porque si las imágenes DPB retrasadas asociadas continúan ocupando la DPB, comienza la decodificación de las imágenes de picA puede volverse problemático, por ejemplo, debido al desbordamiento de la memoria intermedia. En este caso, el valor de no_output_of_prior_pics_flag (por ejemplo, un indicador que indica que las imágenes que fueron previamente decodificadas y almacenadas en la DPB deben eliminarse de la DPB sin salir si se establece en 1, e indica que las imágenes que fueron previamente decodificadas y almacenadas en la DPB no debe eliminarse de la DPB sin salir si se establece en 0) asociado con picA debe establecerse igual a un valor de "1" por el codificador o empalmador, o NoOutputOfPriorPicsFlag (por ejemplo, un valor derivado que puede determinarse basado en la información incluida en el flujo de bits) puede derivarse para que sea igual a un valor de "1" por el decodificador, para eliminar las imágenes retrasadas sin salida de la DPB. Esta operación de empalme se describe con más detalle a continuación con referencia a la **FIG. 4**.

[0136] Este proceso de eliminación de imágenes DPB retrasadas asociadas de la DPB sin salida puede denominarse "liberación". Incluso en situaciones no descritas anteriormente, una imagen IRAP puede especificar el valor de no_output_of_prior_pics_flag igual a un valor de "1", de modo que el decodificador vaciará las imágenes de DPB asociadas de la imagen IRAP.

Flujo de bits incluyendo un punto de empalme

[0137] Con referencia a la **FIG. 4**, se describirá un flujo de bits de ejemplo que tiene un punto de empalme. La **FIG. 4** muestra un flujo de bits 400 de múltiples capas creado mediante el empalme de los flujos de bits 410 y 420. El flujo de bits 410 incluye una capa de mejora (EL) 410A y una capa base (BL) 410B, y el flujo de bits 420 incluye una EL 420A y una BL 420B. La EL 410A incluye una imagen EL 412A, y la BL 410B incluye una imagen BL 412B. La EL 420A incluye imágenes EL 422A, 424A y 426A, y la BL 420B incluye imágenes BL 422B, 424B y 426B. El flujo de bits 400 de múltiples capas incluye además las unidades de acceso (AU) 430-460. La AU 430 incluye la imagen EL 412A y la imagen BL 412B, la AU 440 incluye la imagen EL 422A y la imagen BL 422B, la AU 450 incluye la imagen EL 424A y la imagen BL 424B, y la AU 460 incluye la imagen EL 426A y la imagen BL 426B. En el ejemplo de la **FIG. 4**, la imagen BL 422B es una imagen IRAP, y la imagen EL 422A correspondiente en la AU 440 es una imagen final (por ejemplo, una imagen no IRAP), y en consecuencia, la AU 440 es una AU de IRAP no alineada. Además, debe tenerse en cuenta que el AU 440 es una unidad de acceso que sigue inmediatamente un punto de empalme 470.

[0138] Aunque el ejemplo de la **FIG. 4** ilustra un caso en el que dos flujos de bits diferentes se unen entre sí, en algunos modos de realización, puede estar presente un punto de empalme cuando se elimina una porción del flujo de bits. Por ejemplo, un flujo de bits puede tener porciones A, B y C, la porción B está entre las porciones A y C. Si la porción B se retira del flujo de bits, las porciones restantes A y C pueden unirse, y el punto en el que se unen entre sí puede denominarse un punto de empalme. Más generalmente, un punto de empalme como se discute en la presente solicitud puede considerarse presente cuando uno o más parámetros o indicadores señalizados o derivados tienen valores predeterminados. Por ejemplo, sin recibir una indicación específica de que existe un punto de empalme en

una ubicación particular, un decodificador puede determinar el valor de un indicador (por ejemplo, NoCInrasOutputFlag) y realizar una o más técnicas descritas en esta aplicación en función del valor del indicador.

Liberación de imágenes en contexto multicapa

[0139] El proceso de lavado de imágenes también es pertinente en flujos de bits de múltiples capas. Más específicamente, es pertinente para todas las imágenes que pertenecen a una unidad de acceso IRAP inicial, y también para las imágenes IRAP que no están en una unidad de acceso IRAP inicial. Como se describió anteriormente, en algunas implementaciones existentes como SHVC y MV-HEVC, una unidad de acceso IRAP puede definirse como una unidad de acceso que contiene una imagen IRAP que tiene `nuh_layer_id` igual a un valor de "0" (independientemente de si hay otras imágenes en el la unidad de acceso son imágenes IRAP), y una unidad de acceso IRAP inicial se puede definir como una unidad de acceso que contiene una imagen IRAP que tiene `nuh_layer_id` igual a un valor de "0" y que tiene `NoRasOutputFlag` igual a un valor de "1" (de nuevo independientemente de si otras imágenes en la unidad de acceso son imágenes IRAP).

[0140] Con la posibilidad de tener imágenes IRAP no alineadas en unidades de acceso (por ejemplo, una unidad de acceso puede contener imágenes IRAP e imágenes no IRAP) en SHVC y MV-HEVC, las situaciones descritas en la sección anterior en el contexto de HEVC pueden ocurrir en diferentes capas de un flujo de bits SHVC/MV-HEVC. Por ejemplo, una imagen de imagen CRA que tiene `NoRasOutputFlag` igual a un valor de "1" puede estar presente en una capa de mejora en el medio de un flujo de bits (por ejemplo, no en la primera unidad de acceso del flujo de bits) que comienza con un IRAP inicial unidad de acceso que no tiene una imagen CRA en la misma capa que `picA`. Además, el cambio de resolución de una imagen podría ocurrir en imágenes IRAP en una capa de mejora en una unidad de acceso donde la resolución de la capa base no cambia, o viceversa. Pueden surgir situaciones similares para diferentes tamaños de DPB.

Liberación de imágenes en SVC y MVC

[0141] Debido al diseño de codificación de bucle único de SVC, solo se inserta una imagen reconstruida por unidad de acceso en la DPB, excepto en los casos en que se utiliza la llamada escalabilidad granular media (MGS) (en cuyo caso puede haber dos imágenes decodificadas de las llamadas unidades de acceso de imagen clave que se almacenan en la DPB). Sin embargo, en cada unidad de acceso solo se puede emitir la imagen decodificada de la capa más alta. Las operaciones para gestionar la DPB, incluyendo la liberación de imágenes, por lo tanto, solo se refieren a imágenes en la capa más alta, principalmente porque no se requiere que una imagen decodificada de una capa base esté presente en la DPB para predecir la capa de mejora.

[0142] En MVC, más de una vista puede ser vista de salida de destino, y los componentes de vista decodificados deben mantenerse para predecir los componentes de vista en otra capa, incluso si no son necesarios para predecir los componentes de vista en la misma capa. Por lo tanto, los componentes de vista desde más de una vista pueden estar presentes en la DPB. El indicador `no_output_of_prior_pics_flag` se señala para cada componente de vista IDR (por ejemplo, un componente de vista IDR de una vista no base se señala con `non_idr_flag` igual a un valor de "0"), y la liberación de los componentes de vista es específico de la capa (o específico de la vista). En MVC, por simplicidad, los componentes de la vista IDR en una unidad de acceso en MVC están alineados. Por ejemplo, si un componente de vista en una unidad de acceso es un componente de vista IDR, todos los componentes de vista en esa unidad de acceso también son componentes de vista IDR. Por lo tanto, la operación de vaciado también se realiza en todas las vistas en el flujo de bits, aunque la operación puede ser específica de la vista/capa.

Liberación de imágenes en SHVC y MV-HEVC

[0143] Cuando se produce una liberación bajo el diseño actual en SHVC y MV-HEVC, todas las imágenes en la DPB se eliminan sin salir (por ejemplo, se muestran). No es posible que las imágenes de solo una capa en el flujo de bits (excepto en el caso trivial cuando solo la capa base está presente en el flujo de bits) se vacíen, por lo tanto, la liberación no es específica de la capa.

Conformidad de temporización de salida

[0144] En algunas implementaciones (por ejemplo, SHVC, MV-HEVC, etc.), como el borrador de trabajo SHVC (WD) 3 y el borrador de trabajo MV-HEVC 5, la salida y eliminación de imágenes de la DPB para conformidad de temporización de salida se realizan como se describe abajo. Las partes relevantes para el proceso de liberación se muestran en *cursiva*. En el siguiente ejemplo, la eliminación de imágenes invocadas es específica de cada capa, como se especifica en la Sección F.13,3.2 de la especificación HEVC.

Tabla 1 - Conformidad de temporización de salida en SHVC WD 3 y MV-HEVC WD 5

<p>C.3.2 Eliminación de imágenes de la DPB</p> <p>La eliminación de imágenes de la DPB antes de la decodificación de la imagen actual (pero después de analizar sintácticamente la cabecera de segmento del primer segmento de la imagen actual) sucede instantáneamente en el momento de la eliminación de la CPB de la primera unidad de decodificación de la unidad de acceso n (que contiene la imagen actual) y procede como se indica a continuación:</p> <ul style="list-style-type: none"> - Se invoca el proceso de decodificación para el RPS como se especifica en la subcláusula 8.3.2. - Cuando la imagen actual es una imagen IRAP con <i>NoRasIOutputFlag</i> igual a 1 que no es la imagen 0, se aplican las siguientes etapas ordenadas: <ol style="list-style-type: none"> 1. La variable <i>NoOutputOfPriorPicsFlag</i> se deriva para el decodificador bajo prueba de la siguiente manera: <ul style="list-style-type: none"> - Si la imagen actual es una imagen CRA, <i>NoOutputOfPriorPicsFlag</i> se establece igual a 1 (independientemente del valor de <i>no_output_of_prior_pics_flag</i>). - De otro modo, si el valor de <i>pic_width_in_luma_samples</i>, <i>pic_height_in_luma_samples</i>, o <i>sps_max_dec_pic_buffering_minus1[HighestTid]</i> derivado del SPS activo es diferente del valor de <i>pic_width_in_luma_samples</i>, <i>pic_height_in_luma_samples</i>, o <i>sps_max_dec_pic_buffering_minus1[HighestTid]</i>, respectivamente, derivado del SPS activo para la imagen anterior, <i>NoOutputOfPriorPicsFlag</i> puede (pero no debe) establecerse a 1 mediante el decodificador bajo prueba, independientemente del valor de <i>no_output_of_prior_pics_flag</i>. <p style="text-align: right;"><i>NOTA - Aunque en estas condiciones se prefiere establecer <i>NoOutputOfPriorPicsFlag</i> igual a <i>no_output_of_prior_pics_flag</i>, el decodificador bajo prueba puede establecer <i>NoOutputOfPriorPicsFlag</i> en 1 en este caso.</i></p> - De otro modo, <i>NoOutputOfPriorPicsFlag</i> se establece igual a <i>no_output_of_prior_pics_flag</i>. 2. El valor de <i>NoOutputOfPriorPicsFlag</i> derivado para el decodificador bajo prueba se aplica para el HRD, de modo que cuando el valor de <i>NoOutputOfPriorPicsFlag</i> es igual a 1, todos los almacenamientos intermedios de almacenamiento de imágenes en la DPB se vacían sin salida de las imágenes que contienen, y la plenitud de la DPB se establece igual a 0. - Cuando se cumplen las dos condiciones siguientes para cualquier imagen k en la DPB, todas esas imágenes k en la DPB se eliminan de la DPB: <ul style="list-style-type: none"> - la imagen k está marcada como "no usada como referencia" - la imagen k tiene <i>PicOutputFlag</i> igual a 0 o su tiempo de salida DPB es menor o igual que el tiempo de eliminación de CPB de la primera unidad de decodificación (indicada como unidad de decodificación m) de la imagen actual n; es decir, <i>DpbOutputTime</i> [k] es menor o igual que <i>CpbRemovalTime</i> (m) - Para cada imagen que se elimina de la DPB, la completitud de DPB se reduce en uno. <p>F.13.3.2 Eliminación de imágenes de la DPB</p> <p>Las especificaciones de la subcláusula C.3.2 se aplican por separado para cada conjunto de imágenes decodificadas con un valor particular de <i>nuh_layer_id</i> con las siguientes modificaciones.</p> <ul style="list-style-type: none"> - Reemplazar "La eliminación de imágenes de la DPB antes de la decodificación de la imagen actual (pero después de analizar el encabezado del segmento del primer segmento de la imagen actual) se produce instantáneamente en el momento de eliminación de la CPB de la primera unidad de decodificación de la unidad de acceso n (que contiene el imagen actual) y procede de la siguiente manera:" con "La eliminación de imágenes de la DPB antes de decodificar la imagen actual (pero después de analizar el encabezado de la primera porción de la imagen actual) se produce instantáneamente en el momento de la eliminación de la CPB de la primera unidad de decodificación de la imagen n y procede de la siguiente manera:".
--

Conformidad de orden de salida

5 [0145] En algunas implementaciones (por ejemplo, SHVC, MV-HEVC, etc.), la salida y eliminación de imágenes de la DPB para el cumplimiento del orden de salida se realiza como se describe a continuación. Las partes relevantes para el proceso de liberación se muestran en *cursiva*. En el ejemplo a continuación, la eliminación de imágenes, cuando se invoca, se realiza para todas las capas.

Tabla 2 - Conformidad del orden de salida en SHVC WD 3 y MV-HEVC WD 5

F.13.5.2.2 Salida y eliminación de imágenes de la DPB

La provisión y eliminación de imágenes de la DPB antes de la decodificación de la imagen actual (pero después de analizar sintácticamente la cabecera de segmento del primer segmento de la imagen actual) puede producirse instantáneamente cuando la primera unidad de decodificación de la imagen actual se elimina de la CPB y procede como se indica a continuación. Se invoca el proceso de decodificación para el RPS como se especifica en la subcláusula F.8.3.2.

- Si la imagen actual es una imagen IRAP con *NoRasOutputFlag* igual a 1 y con *nuh_layer_id* igual a 0 que no es la imagen 0, se aplican las siguientes etapas ordenadas:

1. La variable *NoOutputOfPriorPicsFlag* se deriva para el decodificador bajo prueba de la siguiente manera:

- Si la imagen actual es una imagen CRA, *NoOutputOfPriorPicsFlag* se establece igual a 1 (independientemente del valor de *no_output_of_prior_pics_flag*).

- De otro modo, si el valor de *pic_width_in_luma_samples*, *pic_height_in_luma_samples*, o *sps_max_dec_pic_buffering_minus1[HighestTid]* derivado del SPS activo es diferente del valor de *pic_width_in_luma_samples*, *pic_height_in_luma_samples*, o *sps_max_dec_pic_buffering_minus1[HighestTid]*, respectivamente, derivado del SPS activo para la imagen anterior, *NoOutputOfPriorPicsFlag* puede (pero no debe) establecerse a 1 mediante el decodificador bajo prueba, independientemente del valor de *no_output_of_prior_pics_flag*.

NOTA: aunque en estas condiciones se prefiere establecer indicador de no emisión de imagen previa igual a *no_output_of_prior_pics_flag*, el decodificador bajo prueba puede establecer indicador de no emisión de imagen previa en 1 en este caso.

- De otro modo, *NoOutputOfPriorPicsFlag* se establece igual a *no_output_of_prior_pics_flag*.

2. El valor de *NoOutputOfPriorPicsFlag* derivado para el decodificador bajo prueba se aplica para el HRD de la siguiente manera:

- Si *NoOutputOfPriorPicsFlag* es igual a 1, todos los almacenamientos intermedios de almacenamiento de imágenes en la DPB se vacían sin salida de las imágenes que contienen, y la plenitud de la DPB se establece en 0.

- De lo contrario (*NoOutputOfPriorPicsFlag* es igual a 0), todas las memorias intermedias de almacenamiento de imágenes que contienen una imagen que está marcada como "no necesaria para la salida" y "no utilizada para referencia" se vacían (sin salida), y todos los almacenamientos intermedios de almacenamiento de imágenes no vacías en la DPB se vacían invocando repetidamente el proceso de "salto" especificado en la subcláusula F.13.5.2.4, y la plenitud de DPB se establece igual a 0.

- De lo contrario (la imagen actual no es una imagen IRAP con *NoRasOutputFlag* igual a 1 o con *nuh_layer_id* no igual a 0), se vacían todos las memorias intermedias de almacenamiento de imágenes que contienen una imagen marcada como "no necesaria para la salida" y "no utilizada para referencia" (sin salida). Para cada memoria intermedia de almacenamiento de imágenes que se vacía, la plenitud de DPB se reduce en uno. La variable *currLayerId* se establece igual a *nuh_layer_id* de la imagen decodificada actual y cuando una o más de las siguientes condiciones son verdaderas, el proceso de "salto" especificado en la subcláusula F.13.5.2.4 se invoca repetidamente mientras disminuye aún más la plenitud de DPB por uno para cada memoria intermedia de almacenamiento de imágenes adicional que se vacía, hasta que ninguna de las siguientes condiciones sea verdadera:

- El número de imágenes con *nuh_layer_id* igual a *currLayerId* en la DPB que están marcadas como "necesarias para la salida" es mayor que *sps_max_num_reorder_pics [HighestTid]* desde el SPS activo (cuando *currLayerId* es igual a 0) o desde el SPS de la capa activa para el valor de *currLayerId* (cuando *currLayerId* no es igual a 0).

- *sps_max_latency_increase_plus1 [HighestTid]* del SPS activo (cuando *currLayerId* es igual a 0) o el SPS de la capa activa para el valor de *currLayerId* no es igual a 0 y hay al menos una imagen con *nuh_layer_id* igual a *currLayerId* en la DPB que está marcado como "Necesario para la salida" para el cual la variable asociada *PicLatencyCount [currLayerId]* es mayor o igual que *SpsMaxLatencyPictures [HighestTid]* derivada del SPS activo (cuando *currLayerId* es igual a 0) o de la capa activa SPS para el valor de *currLayerId*.

F.13.5.2.2 Salida y eliminación de imágenes de la DPB

- El número de imágenes con `nuh_layer_id` igual a `currLayerId` en la DPB es mayor o igual que `sps_max_dec_pic_buffering_minus1 [HighestTid] + 1` desde el SPS activo (cuando `currLayerId` es igual a 0) o desde el SPS de la capa activa por el valor de `currLayerId`.

Comparación de conformidad de temporización de salida y conformidad de orden de salida

5 [0146] Como se describió anteriormente, la conformidad de temporización de salida y la conformidad del orden de salida pueden no resultar en el mismo comportamiento de descarga. Por ejemplo, para los decodificadores de conformidad de temporización de salida, se invoca el vaciado para cada imagen en una capa que no es la primera imagen de la capa en el flujo de bits y que tiene `NoRasOutputFlag` igual a un valor de "1". Cuando se invoca el vaciado, todas las imágenes decodificadas de esa capa en la DPB son liberadas. Por otro lado, para decodificadores de conformidad de orden de salida, el vaciado solo se invoca para una imagen en la capa base que no es la primera imagen en el flujo de bits y que tiene `NoRasOutputFlag` igual a un valor de "1". Cuando se invoca la liberación, se liberan todas las imágenes decodificadas de todas las capas en la DPB.

15 [0147] En un flujo de bits que tiene dos capas, cuando una imagen LIP en la EL que es una imagen IRAP y no pertenece a una AU IRAP activa una resolución diferente, y la resolución de la BL no puede cambiar en esta AU debido a que la imagen BL está en una AU no IRAP (por ejemplo, no una imagen IRAP), se puede desear una liberación de imágenes específico de la capa. Aquí, solo se deben liberar las imágenes de la EL, pero no de la BL. Esta característica no está disponible para conformidad de orden de salida.

20 [0148] En un flujo de bits que tiene dos capas, en un caso en el que una unidad de acceso incluye una imagen BL que es una imagen IDR y una imagen EL que no es IRAP, la resolución de la imagen BL puede actualizarse en la unidad de acceso, mientras que la resolución de la imagen EL no se actualiza. En tal caso, la liberación debe realizarse solo para las imágenes del BL, y las imágenes EL no deben liberarse. Esta característica no está disponible para conformidad de orden de salida.

Señalización del indicador que indica la salida de imágenes anteriores

25 [0149] En algunas realizaciones, la variable `NoOutputOfPriorPicsFlag` (por ejemplo, un valor derivado por el decodificador para determinar, al decodificar una imagen IRAP, si se emiten o no las imágenes en la DPB antes de que se vacíe la DPB) se basa en `no_output_of_prior_pics_flag` y otras condiciones. Por ejemplo, `no_output_of_prior_pics_flag` puede ser un valor que se señala en el flujo de bits, mientras que `NoOutputOfPriorPicsFlag` puede ser un valor derivado por un codificador basado en la información incluida en el flujo de bits. Un decodificador puede derivar el valor de `NoOutputOfPriorPicsFlag` en función del valor de `no_output_of_prior_pics_flag` y otras condiciones, y luego usar el valor derivado de `NoOutputOfPriorPicsFlag` para determinar si se emiten imágenes o no. En algunas realizaciones, el indicador `NoOutputOfPriorPicsFlag` puede indicar si la unidad de acceso actual comprende un punto de empalme, en el que se unen dos flujos de bits diferentes.

30 [0150] En algunas realizaciones, `NoCIRasOutputFlag` y `NoRasOutputFlag` pueden ser variables derivadas basadas en la información incluida en el flujo de bits. Por ejemplo, `NoRasOutputFlag` puede derivarse para cada imagen IRAP (por ejemplo, en BL y/o EL), y `NoCIRasOutputFlag` puede derivarse solo para las imágenes de capa más baja (por ejemplo, imágenes BL). El valor de cada uno de `NoCIRasOutputFlag` y `NoRasOutputFlag` puede indicar que algunas imágenes en el flujo de bits pueden no ser decodificables correctamente debido a la falta de disponibilidad de ciertas imágenes de referencia. Tal falta de disponibilidad de imágenes de referencia puede producirse en puntos de acceso aleatorio. Las imágenes de salto aleatorio de capas cruzadas (CL-RAS) son, de alguna manera, el equivalente de varias capas de las imágenes RASL. Si un decodificador comienza a decodificar un flujo de bits en un punto de acceso aleatorio (por ejemplo, una unidad de acceso que tiene una imagen BL IRAP), y la imagen EL en la unidad de acceso no es una imagen IRAP, entonces esa imagen EL es una imagen CL-RAS. Todas las imágenes en la EL pueden ser imágenes CL-RAS (por ejemplo, decodificables, pero no correctamente decodificables) hasta que se produzca una imagen IRAP en la EL. Cuando se proporciona dicha imagen EL IRAP en el flujo de bits, se puede decir que EL se ha inicializado.

35 [0151] Por ejemplo, en el ejemplo de la **FIG. 4**, la imagen EL 422A puede ser una imagen LIP que no es una imagen IRAP, y la imagen BL 422B puede ser una imagen IRAP que tiene un indicador `NoCIRasOutputFlag` asociado a la misma. En este ejemplo, el valor de `NoOutputOfPriorPicsFlag` asociado con la imagen EL 422A puede inferirse en función del valor de `NoCIRasOutputFlag` asociado con la imagen BL 422B. Por ejemplo, si `NoCIRasOutputFlag` es igual a un valor de "1", `NoOutputOfPriorPicsFlag` para la imagen EL 422A también se puede establecer en un valor de "1", lo que hace que las imágenes en la DPB no se impriman antes de que se eliminen de la DPB. Por otro lado, si `NoCIRasOutputFlag` es igual a un valor de "0", `NoOutputOfPriorPicsFlag` para la imagen EL 422A también se puede establecer en un valor de "0", lo que hace que las imágenes en la DPB se eliminen de la DPB después de la salida.

Inferencia mejorada de NoOutputOfPriorPicsFlag

[0152] Como se explicó anteriormente, las imágenes de puntos de acceso intra aleatorio (IRAP) pueden proporcionar puntos de acceso aleatorio para decodificar un flujo de bits. Un decodificador puede comenzar a decodificar un flujo de bits decodificando una imagen IRAP sin tener que decodificar imágenes que preceden a la imagen IRAP. En el momento de decodificar una imagen IRAP, la memoria intermedia de imagen decodificada (DPB) puede tener varias imágenes decodificadas en la memoria intermedia. Las imágenes que han sido decodificadas desde un flujo de bits se almacenan en la DPB. Las imágenes decodificadas en la DPB pueden incluir imágenes que están marcadas como "necesarias para la salida" o imágenes que están marcadas como "utilizadas como referencia". Las imágenes decodificadas en la DPB también pueden incluir imágenes que no están marcadas como "necesarias para la salida" ni como "utilizadas para referencia"; tales imágenes decodificadas están presentes en la DPB hasta que son eliminadas por el proceso de decodificación. Si la salida de las imágenes existentes en la DPB afectaría el rendimiento del decodificador (por ejemplo, existen demasiadas imágenes en la DPB para que el decodificador pueda emitirlas), puede ser conveniente eliminar esas imágenes existentes sin emitirlas (por ejemplo, imágenes). No eliminar las imágenes existentes en la DPB según sea necesario puede provocar un desbordamiento de la memoria intermedia o un retraso en la decodificación y salida de imágenes posteriores.

[0153] El `NoOutputOfPriorPicsFlag` variable puede indicar, al decodificar una imagen IRAP, si las imágenes en la DPB deben enviarse antes de ser eliminadas de la DPB. Al decodificar una imagen IRAP, el valor de `NoOutputOfPriorPicsFlag` se puede establecer en 1 cuando las imágenes en la DPB no se deben emitir antes de eliminarlas. De forma alternativa, al decodificar una imagen IRAP, el valor de `NoOutputOfPriorPicsFlag` se puede establecer en 0 cuando las imágenes en la DPB se deben emitir antes de eliminarlas. El valor de `NoOutputOfPriorPicsFlag` puede determinarse en función de un elemento de sintaxis correspondiente y/o diversas condiciones e información. El elemento de sintaxis `no_output_of_prior_pics_flag` puede recibirse en el flujo de bits, y el decodificador puede establecer el valor de `NoOutputOfPriorPicsFlag` igual al valor decodificado de `no_output_of_prior_pics_flag`. O el valor de `NoOutputOfPriorPicsFlag` se puede derivar o inferir en función de diversas condiciones. Por ejemplo, en versiones anteriores de SHVC y MV-HEVC (por ejemplo, borrador de trabajo SHVC 4 y borrador de trabajo MV-HEVC 6), el `NoOutputOfPriorPicsFlag` puede establecerse igual a 1 cuando la resolución espacial o el número de almacenamientos de imágenes cambia. Al establecer `NoOutputOfPriorPicsFlag` en 1, el decodificador puede controlar la cantidad de memoria disponible en la DPB eliminando las imágenes en la DPB según corresponda. Sin embargo, la cantidad de memoria necesaria en la DPB también puede verse afectada por cambios en el formato de color y/o la profundidad de bits de las imágenes. A menos que se suponga el peor de los casos o el formato de color máximo o la profundidad de bits, no considerar el formato de color y/o la profundidad de bits puede conducir a un tamaño de memoria subóptimo para la DPB.

[0154] Para abordar estos y otros desafíos, las técnicas de acuerdo con ciertos aspectos pueden tener en cuenta el formato de color y/o la profundidad de bits de las imágenes al determinar el valor de `NoOutputOfPriorPicsFlag` para las imágenes IRAP. Por ejemplo, las técnicas pueden determinar si el formato de color y/o la profundidad de bits de la imagen actual en la capa actual a decodificar son diferentes del formato de color y/o la profundidad de bits de la imagen anterior en la capa actual. El formato de color y/o la profundidad de bits de la imagen actual y la imagen anterior se pueden obtener del conjunto de parámetros de secuencia actualmente activos (SPS) y el SPS que estaba activo cuando se decodificó la imagen anterior, respectivamente. La profundidad de bits de una imagen puede incluir una profundidad de bits para el componente luma de la imagen y una profundidad de bits para el componente de croma de la imagen. Si hay cambios en el formato de color o en la profundidad de bits, el valor de `NoOutputOfPriorPicsFlag` puede establecerse igual a 1 para activar la eliminación de imágenes en la DPB sin generar las imágenes. En un ejemplo, si la profundidad de bits de las muestras de luma cambia de 8 bits a 16 bits, sería deseable liberar las imágenes en la DPB para que haya más memoria disponible en la DPB.

[0155] Al tener en cuenta los cambios en el formato de color y/o la profundidad de bits de las imágenes, las técnicas pueden determinar el valor de `NoOutputOfPriorPicsFlag` con mayor precisión, y se puede indicar un valor óptimo del tamaño de la DPB que es necesario y suficiente para la decodificación de un flujo de bits. Por ejemplo, las técnicas pueden anticipar mejor situaciones que requerirían memoria adicional en la memoria intermedia. Además de los cambios en el número de almacenamientos de imágenes y la resolución espacial, la DPB se libera para que haya más almacenamiento disponible cuando cambia el formato de color o la profundidad de bits.

[0156] Ciertos detalles relacionados con la inferencia de `NoOutputOfPriorPicsFlag` se explican más adelante. Varios términos utilizados a lo largo de esta divulgación son términos amplios que tienen su significado corriente.

Ejemplo de realización

[0157] Las técnicas mencionadas anteriormente se pueden implementar como se muestra en el siguiente ejemplo. El ejemplo se proporciona en el contexto de versiones anteriores de SHVC y MV-HEVC (por ejemplo, SHVC WD 4 y MV-HEVC WD 6). Los cambios de las versiones anteriores de SHVC y MV-HEVC se indican en cursiva. La sección C.3.2 describe las operaciones DPB de temporización de salida en la eliminación de imágenes en SHVC WD 4 y MV-HEVC WD 6. La sección C.5.2.2 describe el orden de salida de las operaciones DPB en la eliminación de imágenes en SHVC WD 4 y MV-HEVC WD 6. La primera rama "De otro modo" en cada sección se relaciona con la inferencia de `NoOutputOfPriorPicsFlag` basada en el formato de croma y la profundidad de bits de las imágenes.

Tabla 3 - Modo de realización de ejemplo

C.3.2 Eliminación de imágenes de la DPB

Cuando la imagen actual no es la imagen 0 en la capa actual, la eliminación de imágenes en la capa actual de la DPB antes de la decodificación de la imagen actual, es decir, la imagen n, pero después de analizar el encabezado del primer segmento de la imagen actual, sucede instantáneamente en el momento de eliminación de CPB de la primera unidad de decodificación de la imagen actual y procede de la siguiente manera:

- Se invoca el proceso de descodificación para el RPS como se especifica en la subcláusula 8.3.2.
- Cuando la imagen actual es una imagen IRAP con NoRaslOutputFlag igual a 1, o la imagen de la capa base en la unidad de acceso actual es una imagen IRAP con NoRaslOutputFlag igual a 1 y NoCInrasOutputFlag es igual a 1, se aplican las siguientes etapas ordenadas:
 1. La variable NoOutputOfPriorPicsFlag se deriva para el decodificador bajo prueba de la siguiente manera:
 - Si la imagen actual es una imagen CRA con NoRaslOutputFlag igual a 1, NoOutputOfPriorPicsFlag se establece igual a 1 (independientemente del valor de no_output_of_prior_pics_flag).
 - De lo contrario, si la imagen actual es una imagen IRAP con NoRaslOutputFlag igual a 1 y el valor de pic_width_in_luma_samples, pic_height_in_luma_samples, chroma_format_idc, bit_depth_luma_minus8, bit_depth_chroma_minus8, or sps_max_dec_pic_buffering_minus1[HighestTid] derivado del SPS activo para la capa actual es diferente del valor de pic_width_in_luma_samples, pic_height_in_luma_samples, chroma_format_idc, bit_depth_luma_minus8, bit_depth_chroma_minus8, or sps_max_dec_pic_buffering_minus1[HighestTid], respectivamente, derivados del SPS que estaba activo para la capa actual cuando se decodifica la imagen anterior en la capa actual, NoOutputOfPriorPicsFlag puede (pero no debe) establecerse en 1 mediante el decodificador bajo prueba, independientemente del valor de no_output_of_prior_pics_flag.

NOTA - Aunque en estas condiciones se prefiere establecer NoOutputOfPriorPicsFlag igual a no_output_of_prior_pics_flag, el decodificador bajo prueba puede establecer NoOutputOfPriorPicsFlag en 1 en este caso.
 - De lo contrario, si la imagen actual es una imagen IRAP con NoRaslOutputFlag igual a 1, NoOutputOfPriorPicsFlag se establece igual a no_output_of_prior_pics_flag.
 - De lo contrario (la imagen actual no es una imagen IRAP con NoRaslOutputFlag igual a 1, la imagen de la capa base en la unidad de acceso actual es una imagen IRAP con NoRaslOutputFlag igual a 1 y NoCInrasOutputFlag es igual a 1), NoOutputOfPriorPicsFlag se establece igual a 1.
 2. El valor de NoOutputOfPriorPicsFlag derivado para el decodificador bajo prueba se aplica para el HRD, de modo que cuando el valor de NoOutputOfPriorPicsFlag es igual a 1, todos los almacenamientos intermedios de almacenamiento de imágenes en la DPB se vacían sin salida de las imágenes que contienen, y la plenitud de la DPB se establece igual a 0.
- Cuando se cumplen las dos condiciones siguientes para cualquier imagen k en la DPB, todas esas imágenes k en la DPB se eliminan de la DPB:
 - la imagen k está marcada como "no usada como referencia"
 - la imagen k tiene PicOutputFlag igual a 0 o su tiempo de salida DPB es menor o igual que el tiempo de eliminación de CPB de la primera unidad de decodificación (indicada como unidad de decodificación m) de la imagen actual n; es decir, DpbOutputTime [k] es menor o igual que CpbRemovalTime (m)
- Para cada imagen que se elimina de la DPB, la completitud de DPB se reduce en uno.

C.5.2.2 Salida y eliminación de imágenes de la DPB

Cuando la imagen actual no es la imagen 0 en la capa actual, la salida y eliminación de imágenes en la capa actual de la DPB antes de la decodificación de la imagen actual, es decir, la imagen n, pero después de analizar el encabezado de la primera porción de la capa de la imagen actual, ocurre instantáneamente cuando la primera unidad de decodificación de la imagen actual se elimina de la CPB y procede de la siguiente manera:

- Se invoca el proceso de descodificación para el RPS como se especifica en la subcláusula 8,3.2.

C.5.2.2 Salida y eliminación de imágenes de la DPB

- Si la imagen actual es una imagen IRAP con NoRasOutputFlag igual a 1, o la imagen de la capa base en la unidad de acceso actual es una imagen IRAP con NoRasOutputFlag igual a 1 y NoCInrasOutputFlag es igual a 1, se aplican las siguientes etapas ordenadas:
 1. La variable NoOutputOfPriorPicsFlag se deriva para el decodificador bajo prueba de la siguiente manera:
 - Si la imagen actual es una imagen CRA con NoRasOutputFlag igual a 1, NoOutputOfPriorPicsFlag se establece igual a 1 (independientemente del valor de no_output_of_prior_pics_flag).
 - De lo contrario, si la imagen actual es una imagen IRAP con NoRasOutputFlag igual a 1 y el valor de pic_width_in_luma_samples, pic_height_in_luma_samples, *chroma_format_idc*, *bit_depth_luma_minus8*, *bit_depth_chroma_minus8*, or *sps_max_dec_pic_buffering_minus1*[HighestTid] derivado del SPS activo para la capa actual es diferente del valor de pic_width_in_luma_samples, pic_height_in_luma_samples, *chroma_format_idc*, *bit_depth_luma_minus8*, *bit_depth_chroma_minus8*, or *sps_max_dec_pic_buffering_minus1*[HighestTid], respectivamente, derivados del SPS que estaba activo para la capa actual cuando se decodifica la imagen anterior en la capa actual, NoOutputOfPriorPicsFlag puede (pero no debe) establecerse en 1 mediante el decodificador bajo prueba, independientemente del valor de no_output_of_prior_pics_flag.

NOTA - Aunque en estas condiciones se prefiere establecer NoOutputOfPriorPicsFlag igual a no_output_of_prior_pics_flag, el decodificador bajo prueba puede establecer NoOutputOfPriorPicsFlag en 1 en este caso.
 - De lo contrario, si la imagen actual es una imagen IRAP con NoRasOutputFlag igual a 1, NoOutputOfPriorPicsFlag se establece igual a no_output_of_prior_pics_flag.
 - De lo contrario (la imagen actual no es una imagen IRAP con NoRasOutputFlag igual a 1, la imagen de la capa base en la unidad de acceso actual es una imagen IRAP con NoRasOutputFlag igual a 1 y NoCInrasOutputFlag es igual a 1), NoOutputOfPriorPicsFlag se establece igual a 1.
 2. El valor de NoOutputOfPriorPicsFlag derivado para el decodificador bajo prueba se aplica para el HRD de la siguiente manera:
 - Si NoOutputOfPriorPicsFlag es igual a 1, todos los almacenamientos intermedios de almacenamiento de imágenes en la sub-DPB se vacían sin salida de las imágenes que contienen, y la plenitud de la sub-DPB se establece en 0.
 - De lo contrario (NoOutputOfPriorPicsFlag es igual a 0), todas las memorias intermedias de almacenamiento de imágenes que contienen una imagen que está marcada como "no necesaria para la salida" y "no utilizada para referencia" se vacían (sin salida), y todos los almacenamientos intermedios de almacenamiento de imágenes no vacías en la sub-DPB se vacían invocando repetidamente el proceso de "salto" especificado en la subcláusula C.5.2.4, y la plenitud de sub-DPB se establece igual a 0.
- De lo contrario, todos los almacenamientos intermedios de almacenamiento de imágenes que contienen una imagen en la capa actual y que están marcados como "no necesarios para la salida" y "no utilizados para referencia" se vacían (sin salida). Para cada memoria intermedia de almacenamiento de imágenes que se vacía, la plenitud de sub-DPB se reduce en uno. Cuando una o más de las siguientes condiciones son verdaderas, el proceso de "salto" especificado en la subcláusula C.5.2.4 se invoca repetidamente mientras disminuye aún más la plenitud de sub-DPB por uno para cada memoria intermedia de almacenamiento de imágenes adicional que se vacía, hasta que ninguna de las siguientes condiciones sea verdadera:
 - El número de unidades de acceso que contienen al menos una imagen decodificada en la DPB marcado como "necesario para la salida" es mayor que MaxNumReorderPics.
 - MaxLatencyIncreasePlus1 no es igual a 0 y hay al menos una unidad de acceso que contiene al menos una imagen decodificada en la DPB marcada como "necesaria para la salida" para la cual la variable asociada PicLatencyCount es mayor o igual que MaxLatencyPictures.

C.5.2.2 Salida y eliminación de imágenes de la DPB

- El número de imágenes en la capa actual en la sub-DPB es mayor o igual que $\text{MaxDecPicBufferingMinus1} + 1$.

[0158] Según algunas realizaciones, los elementos y las variables de sintaxis utilizados en la inferencia de `NoOutputOfPriorPicsFlag` pueden referirse a lo siguiente:

- 5 • El elemento de sintaxis `pic_width_in_luma_samples` se refiere al ancho de una imagen en unidades de muestras de luma.
- La altura de la imagen del elemento de sintaxis en las muestras de luma se refiere a la altura de una imagen en unidades de muestras de luma.
- 10 • El elemento de sintaxis `chroma_format_idc` se refiere al formato de croma de una imagen.
- El elemento de sintaxis `bit_depth_luma_minus8` se refiere a la profundidad de bits de las muestras de luma de una imagen menos 8 bits.
- 15 • El elemento de sintaxis `bit_depth_chroma_minus8` se refiere a la profundidad de bits de las muestras de croma de una imagen menos 8 bits.
- El elemento de sintaxis `sps_max_dec_pic_buffering_minus1 [HighestTid]` se refiere al número de almacenes de imágenes que se requieren para decodificar una capa de una secuencia de video codificada.
- 20

[0159] Los elementos de sintaxis anteriores se pueden derivar del SPS activo para la capa actual en relación con la imagen actual de la capa actual que se está decodificando. Los elementos de sintaxis anteriores también se pueden derivar en relación con la imagen anterior de la capa actual del SPS que estaba activo para la capa actual cuando se decodificó la imagen anterior. La imagen anterior puede ser una imagen en la capa actual que precede a la imagen actual en orden de decodificación.

[0160] Si una imagen es una imagen IRAP y el valor de `NoRaslOutputFlag` es igual a 1 para la imagen IRAP, el valor de `NoOutputOfPriorPicsFlag` puede establecerse en 1 si los valores de los elementos de sintaxis anteriores para la imagen actual difieren de los valores de la sintaxis respectiva elementos para la imagen anterior. La variable `NoRaslOutputFlag` puede indicar si las imágenes iniciales omitidas de acceso aleatorio (RASL) se emiten en asociación con la imagen actual. Si `NoRaslOutputFlag` es igual a 1, algunas imágenes de referencia a las que se refieren las imágenes RASL no estarán presentes entre predicciones y, por lo tanto, las imágenes RASL no se deben emitir/mostrar. Como se explicó anteriormente, las imágenes RASL pueden referirse a imágenes principales asociadas con una imagen IRAP que no se deberían emitir/mostrar si la decodificación del flujo de bits comienza desde esa imagen IRAP. El decodificador puede comparar cada elemento de sintaxis de la imagen actual con el elemento de sintaxis correspondiente de la imagen anterior, y si el valor de cualquier elemento de sintaxis ha cambiado, `NoOutputOfPriorPicsFlag` puede establecerse en 1, independientemente del valor del elemento de sintaxis `no_output_of_prior_pics_flag`. Como indica el texto de la especificación, puede ser preferente establecer el valor de `NoOutputOfPriorPicsFlag` en el valor del elemento de sintaxis `no_output_of_prior_pics_flag`. Pero el decodificador puede establecer el valor de `NoOutputOfPriorPicsFlag` en 1, si lo desea. Por ejemplo, en algunos casos, el tamaño de DPB es lo suficientemente grande como para que, incluso si el valor del primer indicador se establece como igual al valor descodificado de `no_output_of_prior_pics_flag`, no se produciría un desbordamiento del tamaño de DPB. En tales casos, puede ser beneficioso permitir que el decodificador elija si establecer el valor del primer indicador igual a 1 o igual al valor decodificado de `no_output_of_prior_pics_flag`.

[0161] En un modo de realización, el valor de `chroma_format_idc` cambia de la imagen anterior a la imagen actual. Como se explicó anteriormente, los valores para los elementos de sintaxis se derivan del SPS para la capa actual que está activa al decodificar la imagen actual de la capa actual y el SPS para la capa actual que estaba activa al decodificar la imagen anterior de la capa actual. Cuando el valor de `chroma_format_idc` de la imagen actual y el valor de `chroma_format_idc` de la imagen anterior son diferentes, el decodificador establece el valor de `NoOutputofPriorPicsFlag` en 1, independientemente del valor del elemento de sintaxis `no_output_of_prior_pics_flag`.

[0162] En otro modo de realización, el valor de `bit_depth_luma_minus8` cambia de la imagen anterior a la imagen actual. Cuando el valor de `bit_depth_luma_minus8` de la imagen actual y el valor de `bit_depth_luma_minus8` de la imagen anterior son diferentes, el decodificador establece el valor de `NoOutputofPriorPicsFlag` en 1, independientemente del valor del elemento de sintaxis `no_output_of_prior_pics_flag`.

[0163] En otro modo de realización más, el valor de `bit_depth_chroma_minus8` cambia de la imagen anterior a la imagen actual. Cuando el valor de `bit_depth_chroma_minus8` de la imagen actual y el valor de

bit_depth_chroma_minus8 de la imagen anterior son diferentes, el decodificador establece el valor de NoOutputofPriorPicsFlag en 1, independientemente del valor del elemento de sintaxis no_output_of_prior_pics_flag.

[0164] Generalmente, se necesita más memoria en la DPB cuando el valor de los elementos de sintaxis anteriores aumenta de la imagen anterior a la imagen actual, en lugar de cuando disminuye el valor de los elementos de sintaxis anteriores. Sin embargo, el decodificador puede obtener una cantidad suficiente de memoria en la DPB eliminando imágenes en la DPB sin salida siempre que los valores de los elementos de sintaxis anteriores cambien, ya sea aumentando o disminuyendo. De esta manera, se puede liberar memoria adicional en la DPB usando un paso condicional, en lugar de usar varios pasos para considerar si el valor de los elementos de sintaxis aumentó o disminuyó. Verificar los cambios en los valores de los elementos de sintaxis, en lugar de determinar si los valores aumentaron o disminuyeron, puede hacer que el proceso de inferencia para NoOutputOfPriorPicsFlag sea más eficiente.

Procedimiento para determinar el valor de NoOutputOfPriorPicsFlag

[0165] La FIG. 5 es un diagrama de flujo que ilustra un procedimiento de codificación de información de vídeo, de acuerdo con un modo de realización de la presente divulgación. El procedimiento se relaciona con la determinación del valor de un indicador que indica si las imágenes en la memoria intermedia de imágenes decodificadas deben emitirse (por ejemplo, NoOutputOfPriorPicsFlag). El proceso 500 puede realizarse mediante un codificador (por ejemplo, el codificador como se muestra en la FIG. 2A, 2B, etc.), un decodificador (por ejemplo, el decodificador como se muestra en la FIG. 3A, 3B, etc.) o cualquier otro componente, dependiendo del modo de realización. Los bloques del proceso 500 se describen con respecto al decodificador 33 en la FIG. 3B, pero el proceso 500 puede ser realizado por otros componentes, como un codificador, como se mencionó anteriormente. El decodificador de vídeo de capa 1 30B del decodificador 33 y/o el decodificador de capa 0 30A del decodificador 33 pueden realizar el proceso 500, dependiendo del modo de realización. Todos los modos de realización descritos con respecto a la FIG. 5 pueden implementarse por separado, o en combinación entre sí. Ciertos detalles relacionados con el proceso 500 se explican anteriormente, por ejemplo, con respecto a la FIG. 4.

[0166] El proceso 500 se inicia en el bloque 501. El decodificador 33 puede incluir una memoria (por ejemplo, la memoria intermedia de imagen decodificada 160) para almacenar información de video asociada con una capa actual para decodificar.

[0167] En el bloque 502, el decodificador 33 obtiene el formato de croma de una imagen actual de la capa actual a decodificar, la profundidad de bits de las muestras de croma de la imagen actual y la profundidad de bits de la muestra de luma de la imagen actual. La imagen actual puede ser una imagen de punto de acceso intra aleatorio (IRAP) que inicia una nueva secuencia de video codificada (CVS). El formato de croma de la imagen actual, la profundidad de bits de las muestras de luma de la imagen actual y la profundidad de bits de las muestras de croma de la imagen actual pueden derivarse de un primer conjunto de parámetros de secuencia (SPS) que esté activo para la capa actual en el momento de decodificar la imagen actual.

[0168] Un indicador asociado con la imagen actual que indica que las imágenes iniciales omitidas de acceso aleatorio (RASL) no deben emitirse puede indicar que la imagen actual inicia un nuevo CVS. Si el indicador está habilitado, el indicador indica que la imagen actual inicia un nuevo CVS; si el indicador no está habilitado, el indicador indica que la imagen actual no inicia un nuevo CVS. El indicador puede ser igual o similar a NoRaslOutputFlag. Por ejemplo, NoRaslOutputFlag se puede configurar igual a 1 para indicar que la imagen actual inicia un nuevo CVS.

[0169] En el bloque 503, el decodificador 33 obtiene el formato de croma de la imagen anterior de la capa actual que se decodificó, la profundidad de bits de las muestras de croma de la imagen anterior y la profundidad de bits de las muestras de luma de la imagen anterior. El formato de croma de la imagen anterior, la profundidad de bits de las muestras de luma de la imagen anterior y la profundidad de bits de las muestras de croma de la imagen anterior pueden derivarse de un segundo SPS que estaba activo para la capa actual en el momento de decodificando la imagen anterior.

[0170] En algunos modos de realización, los valores respectivos de la profundidad de bits de las muestras de luma de la imagen actual, la profundidad de bits de las muestras de croma de la imagen actual, la profundidad de bits de las muestras de luma de la imagen anterior y la profundidad de bits de las muestras de croma de la imagen anterior se especifican como 8 bits menos los valores de profundidad de bits reales respectivos. Por ejemplo, la profundidad de bits de las muestras de luma de la imagen actual es proporcionada por el valor del elemento de sintaxis bit_depth_luma_minus8, donde bit_depth_luma_minus8 representa 8 bits menos el valor real de la profundidad de bits de las muestras de luma de la imagen actual.

[0171] En el bloque 504, el decodificador 33 determina el valor de un primer indicador asociado con la imagen actual que indica si las imágenes en una memoria intermedia de imagen decodificada (DPB) deberían emitirse. El decodificador 33 puede determinar el valor del primer indicador en función de: (1) el formato de croma de la imagen actual y el formato de croma de la imagen anterior, (2) la profundidad de bits de las muestras de croma de la imagen actual y la profundidad de bits de las muestras de croma de la imagen anterior, o (3) la profundidad de bits de las

muestras de luma de la imagen actual y la profundidad de bits de las muestras de croma de la imagen anterior. El primer indicador puede ser o similar a NoOutputOfPriorPicsFlag explicado anteriormente.

[0172] En determinados modos de realización, en respuesta a la determinación de que (4) el formato de croma de la imagen actual y el formato de croma de la imagen anterior son diferentes, (5) la profundidad de bits de las muestras de luma de la imagen actual y la profundidad de bits de las muestras de luma de la imagen anterior son diferentes, o (6) la profundidad de bits de las muestras de croma de la imagen actual y la profundidad de bits de las muestras de croma de la imagen anterior son diferentes, el decodificador 33 establece el valor del primer indicador igual a 1. El valor de 1 se proporciona como un ejemplo con fines ilustrativos, pero el decodificador 33 puede usar cualquier valor que indique que el primer indicador está habilitado, según corresponda. Cuando el primer indicador está habilitado, las imágenes en la DPB se eliminan, pero no se imprimen.

[0173] En otros modos de realización, el decodificador 33 decodifica un elemento de sintaxis asociado con la imagen actual en el flujo de bits que indica si se deben emitir imágenes en la DPB. En respuesta a la determinación de que (4) el formato de croma de la imagen actual y el formato de croma de la imagen anterior son diferentes, (5) la profundidad de bits de las muestras de luma de la imagen actual y la profundidad de bits de las muestras de luma de la imagen anterior son diferentes, o (6) la profundidad de bits de las muestras de croma de la imagen actual y la profundidad de bits de las muestras de croma de la imagen anterior son diferentes, el decodificador 33 establece el valor del primer indicador igual al valor decodificado del elemento de sintaxis. Por ejemplo, el elemento de sintaxis es `no_output_of_prior_pics_flag`, e incluso si hay cambios en el formato de croma, la profundidad de bits de luma y la profundidad de bits de croma entre la imagen anterior y la imagen actual, el decodificador 33 establece el valor del primer indicador en el valor descodificado de `no_output_of_prior_pics_flag`.

[0174] En determinados modos de realización, el decodificador 33 obtiene el ancho de la imagen actual, la altura de la imagen actual y el número de almacenes de imágenes asociados con la imagen actual. El ancho de la imagen actual y la altura de la imagen actual se especifican en unidades de muestras de luma, y el número de almacenes de imágenes asociados con la imagen actual indica el tamaño mínimo de DPB requerido para la decodificación de la capa actual en la secuencia de video codificada que comienza desde la imagen actual. El número de almacenes de imágenes se puede especificar para un CVS y señalizarse en el SPS. Para cada CVS y una capa particular, solo un SPS puede estar en vigor. El decodificador 33 también obtiene el ancho de la imagen anterior, la altura de la imagen anterior y el número de almacenes de imágenes asociados con la imagen anterior. El ancho de la imagen anterior y la altura de la imagen anterior se especifican en unidades de muestras de luma, y el número de almacenes de imágenes asociados con la imagen anterior indica el tamaño mínimo de DPB requerido para la decodificación de la capa actual en el final de secuencia de video codificado en la imagen anterior. El ancho de la imagen actual, la altura de la imagen actual y el número de almacenes de imágenes asociados con la imagen actual pueden derivarse de un primer conjunto de parámetros de secuencia (SPS) que esté activo para la capa actual en el momento de decodificar la imagen actual. El ancho de la imagen anterior, la altura de la imagen anterior y el número de almacenes de imágenes asociados con la imagen anterior pueden derivarse de un segundo SPS que estaba activo para la capa actual en el momento de decodificar la imagen anterior.

[0175] En un modo de realización, en respuesta a la determinación de que (7) el ancho de la imagen actual y el ancho de la imagen anterior son diferentes, (8) la altura de la imagen actual y la altura de la imagen anterior son diferentes, o (9) el número de almacenes de imágenes asociados con la imagen actual y el número de almacenes de imágenes asociados con la imagen anterior son diferentes, el decodificador 33 establece el valor del primer indicador igual a 1.

[0176] En otro modo de realización, el decodificador 33 decodifica un elemento de sintaxis asociado con la imagen actual en el flujo de bits que indica si las imágenes en la DPB deberían ser emitidas (por ejemplo, `no_output_of_prior_pics_flag`). En respuesta a la determinación de que (7) el ancho de la imagen actual y el ancho de la imagen anterior son diferentes, (8) la altura de la imagen actual y la altura de la imagen anterior son diferentes, o (9) el número de los almacenes de imágenes asociados con la imagen actual y el número de almacenes de imágenes asociados con la imagen anterior son diferentes, el decodificador 33 establece el valor del primer indicador igual a un valor decodificado del elemento de sintaxis.

[0177] El proceso 500 finaliza en el bloque 505. Los bloques pueden agregarse y/u omitirse en el proceso 500, dependiendo del modo de realización, y los bloques del proceso 500 pueden realizarse en diferentes órdenes, dependiendo del modo de realización.

[0178] Cualquier característica y/o modo de realización descrito con respecto a la interferencia de NoOutputOfPriorPicsFlag en esta divulgación puede implementarse por separado o en cualquier combinación de las mismas. Por ejemplo, cualquier característica y / o realización descrita en relación con las FIGS. 1-4 y otras partes de la divulgación puede implementarse en cualquier combinación con cualquiera de las características y / o realizaciones descritas en relación con la FIG. 5, y viceversa.

[0179] La información y las señales divulgadas en el presente documento pueden representarse usando cualquiera entre varias tecnologías y técnicas diferentes. Por ejemplo, los datos, instrucciones, comandos, información, señales,

bits, símbolos y segmentos que se pueden haber mencionado a lo largo de la descripción anterior se pueden representar mediante tensiones, corrientes, ondas electromagnéticas, campos o partículas magnéticas, campos o partículas ópticas o cualquier combinación de los mismos.

5 **[0180]** Los diversos bloques lógicos, circuitos y etapas de algoritmo ilustrativos, descritos en relación con los modos de realización divulgados en el presente documento pueden implementarse como hardware electrónico, software informático o combinaciones de ambos. Para ilustrar claramente esta intercambiabilidad de hardware y software, anteriormente se han descrito diversos componentes, bloques, módulos, circuitos y etapas ilustrativas, en general, en lo que respecta a su funcionalidad. Que dicha funcionalidad se implemente como hardware o software depende de la aplicación particular y de las restricciones de diseño impuestas en el sistema general. Los expertos en la técnica pueden implementar la funcionalidad descrita de formas distintas para cada solicitud particular, pero no debería interpretarse que dichas decisiones de implementación suponen apartarse del alcance de la presente invención.

15 **[0181]** Las técnicas descritas en el presente documento pueden implementarse en hardware, software, firmware o en cualquier combinación de lo anterior. Dichas técnicas pueden implementarse en cualquiera entre una variedad de dispositivos tales como ordenadores de propósito general, equipos manuales de dispositivos de comunicación inalámbrica o dispositivos de circuitos integrados que tienen múltiples usos, incluyendo su aplicación en equipos manuales de dispositivos de comunicación inalámbrica y otros dispositivos. Todas las características descritas como módulos o componentes pueden implementarse juntas en un dispositivo lógico integrado o por separado, como dispositivos lógicos discretos pero interoperables. Si se implementan en software, las técnicas pueden realizarse, al menos en parte, mediante un medio de almacenamiento de datos legible por ordenador que comprenda código de programa que incluye instrucciones que, cuando se ejecutan, realizan uno o más de los procedimientos descritos anteriormente. El medio de almacenamiento de datos legible por ordenador puede formar parte de un producto de programa informático, que puede incluir materiales de embalaje. El medio legible por ordenador puede comprender memoria o medios de almacenamiento de datos, tales como memoria de acceso aleatorio (RAM), tal como memoria de acceso aleatorio dinámica síncrona (SDRAM), memoria de solo lectura (ROM), memoria de acceso aleatorio no volátil (NVRAM), memoria de solo lectura programable y borrable eléctricamente (EEPROM), memoria FLASH, medios de almacenamiento de datos magnéticos u ópticos, y similares. Las técnicas se pueden realizar, adicionalmente o de forma alternativa, al menos en parte, por un medio de comunicación legible por ordenador que transporta o comunica código de programa en forma de instrucciones o estructuras de datos y a las que se puede acceder, leer y/o ejecutar por medio de un ordenador, tales como señales u ondas propagadas.

35 **[0182]** El código de programa puede ser ejecutado por un procesador, que puede incluir uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados específicos de la aplicación (ASIC), matrices de puertas programables en el terreno (FPGA) u otros circuitos lógicos equivalentes, integrados o discretos. Un procesador de este tipo puede estar configurado para realizar cualquiera de las técnicas descritas en esta divulgación. Un procesador de propósito general puede ser un microprocesador, pero, de forma alternativa, el procesador puede ser cualquier procesador, controlador, microcontrolador o máquina de estados convencional. Un procesador también puede implementarse como una combinación de dispositivos informáticos, por ejemplo, una combinación de un DSP y un microprocesador, una pluralidad de microprocesadores, uno o más microprocesadores junto con un núcleo de DSP o cualquier otra configuración de este tipo. Por consiguiente, el término "procesador", como se usa en el presente documento puede referirse a cualquier estructura anterior, cualquier combinación de la estructura anterior, o cualquier otra estructura o aparato adecuados para la aplicación 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 software o módulos de hardware dedicados configurados para la codificación y la decodificación, o incorporados en un codificador-descodificador de vídeo combinado (CODEC). Además, las técnicas se podrían implementar por completo en uno o más circuitos o elementos lógicos.

50 **[0183]** Las técnicas de esta divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, incluyendo un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). En esta divulgación se describen diversos componentes, módulos o unidades para destacar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no se requiere necesariamente su realización por diferentes unidades de hardware. En su lugar, como se ha descrito anteriormente, diversas unidades se pueden combinar en una unidad de hardware de códec o proporcionar mediante un grupo de unidades de hardware interoperativas, incluidos uno o más procesadores como los descritos anteriormente, junto con software y/o firmware adecuados.

60 **[0184]** Se han descrito diversos modos de realización de la divulgación. La invención se define en las reivindicaciones adjuntas.

REIVINDICACIONES

1. Un aparato para codificar o decodificar información de video, comprendiendo el aparato:

- 5 medios para almacenar información de video asociada con una capa actual;
- medios para obtener al menos uno de los formatos de croma de una imagen actual de la capa actual a decodificar, una profundidad de bits de muestras de luma de la imagen actual, o una profundidad de bits de muestras de croma de la imagen actual, en el que la imagen actual es una imagen de punto de acceso intra aleatorio, IRAP, que inicia una nueva secuencia de video codificada, CVS;
- 10 medios para obtener al menos uno de un formato de croma de una imagen anterior de la capa actual que se decodificó, una profundidad de bits de muestras de luma de la imagen anterior, o una profundidad de bits de muestras de croma de la imagen anterior; y
- 15 medios para determinar un valor de un primer indicador asociado con la imagen actual, indicando el primer indicador si las imágenes previamente decodificadas en una memoria intermedia de imágenes decodificadas, DPB, deben eliminarse de la DPB sin emitirse, las imágenes decodificadas previamente decodificadas de una o más capas, **caracterizado por que**
- 20 la determinación del valor del primer indicador se basa en al menos uno de: el formato de croma de la imagen actual y el formato de croma de la imagen anterior, la profundidad de bits de las muestras de luma de la imagen actual y la profundidad de bits de muestras de luma de la imagen anterior, o la profundidad de bits de las muestras de croma de la imagen actual y la profundidad de bits de las muestras de croma de la imagen anterior.

2. El aparato de la reivindicación 1, en el que los medios para determinar el valor del primer indicador determinan el valor del primer indicador mediante:

- 30 en respuesta a la determinación de que el formato de croma de la imagen actual y el formato de croma de la imagen anterior son diferentes, la profundidad de bits de las muestras de luma de la imagen actual y la profundidad de bits de las muestras de luma de la imagen anterior son diferentes, o la profundidad de bits de las muestras de croma de la imagen actual y la profundidad de bits de las muestras de croma de la imagen anterior son diferentes:
- 35 establecer el valor del primer indicador igual a 1.

3. El aparato de la reivindicación 1, en el que:

- 40 el formato de croma de la imagen actual, la profundidad de bits de las muestras de luma de la imagen actual y la profundidad de bits de las muestras de croma de la imagen actual se derivan de un primer conjunto de parámetros de secuencia, SPS, que esté activo para la capa actual en el momento de decodificar la imagen actual; y
- 45 el formato de croma de la imagen anterior, la profundidad de bits de las muestras de luma de la imagen anterior y la profundidad de bits de las muestras de croma de la imagen anterior se derivan de un segundo SPS que estaba activo para la capa actual en el momento de decodificando la imagen anterior.

4. El aparato para decodificar de la reivindicación 1, en el que el aparato comprende además medios para:

- 50 decodificar un elemento de sintaxis en un flujo de bits que indica si las imágenes en la DPB se deben emitir, en el que el elemento de sintaxis está asociado con la imagen actual; y
- determinar el valor del primer indicador mediante:
- 55 en respuesta a la determinación de que el formato de croma de la imagen actual y el formato de croma de la imagen anterior son diferentes, la profundidad de bits de las muestras de luma de la imagen actual y la profundidad de bits de las muestras de luma de la imagen anterior son diferentes, o la profundidad de bits de las muestras de croma de la imagen actual y la profundidad de bits de las muestras de croma de la imagen anterior son diferentes:
- 60 establecer el valor del primer indicador igual a un valor descodificado del elemento de sintaxis.

5. Un procedimiento de codificación o decodificación de información de vídeo, comprendiendo el procedimiento:

- 65 almacenar información asociada con un flujo de bits de video que incluye una o más capas, cada una de las cuales representa una combinación particular de una o más de: calidad, velocidad de bits, velocidad de tramas, resolución espacial o punto de vista, incluyendo una o más capas una capa actual;

5 obtener al menos uno de los formatos de croma de una imagen actual de la capa actual a decodificar, una profundidad de bits de muestras de luma de la imagen actual, o una profundidad de bits de muestras de croma de la imagen actual, en el que la imagen actual es una imagen de punto de acceso intra aleatorio, IRAP, que inicia una nueva secuencia de video codificada, CVS;

10 obtener al menos uno de un formato de croma de una imagen anterior de la capa actual que se decodificó, una profundidad de bits de muestras de luma de la imagen anterior, o una profundidad de bits de muestras de croma de la imagen anterior; y

15 determinar un valor de un primer indicador asociado con la imagen actual, indicando el primer indicador si las imágenes previamente decodificadas en una memoria intermedia de imágenes decodificadas, DPB, deben eliminarse de la DPB sin emitirse, las imágenes decodificadas previamente decodificadas de una o más capas,

caracterizado por que

20 la determinación del valor del primer indicador se basa en al menos uno de: el formato de croma de la imagen actual y el formato de croma de la imagen anterior, la profundidad de bits de las muestras de luma de la imagen actual y la profundidad de bits de muestras de luma de la imagen anterior, o la profundidad de bits de las muestras de croma de la imagen actual y la profundidad de bits de las muestras de croma de la imagen anterior.

6. El procedimiento de la reivindicación 5, en el que determinar el valor del primer indicador comprende:

25 en respuesta a la determinación de que el formato de croma de la imagen actual y el formato de croma de la imagen anterior son diferentes, la profundidad de bits de las muestras de luma de la imagen actual y la profundidad de bits de las muestras de luma de la imagen anterior son diferentes, o la profundidad de bits de las muestras de croma de la imagen actual y la profundidad de bits de las muestras de croma de la imagen anterior son diferentes:

30 establecer el valor del primer indicador igual a 1.

7. El procedimiento de decodificación de la reivindicación 5, que comprende además:

35 decodificar un elemento de sintaxis en un flujo de bits que indica si las imágenes en la DPB se deben emitir, en el que el elemento de sintaxis está asociado con la imagen actual; y

determinar el valor del primer indicador mediante:

40 en respuesta a la determinación de que el formato de croma de la imagen actual y el formato de croma de la imagen anterior son diferentes, la profundidad de bits de las muestras de luma de la imagen actual y la profundidad de bits de las muestras de luma de la imagen anterior son diferentes, o la profundidad de bits de las muestras de croma de la imagen actual y la profundidad de bits de las muestras de croma de la imagen anterior son diferentes:

45 establecer el valor del primer indicador igual a un valor descodificado del elemento de sintaxis.

8. El procedimiento según la Reivindicación 5, en el que:

50 el formato de croma de la imagen actual, la profundidad de bits de las muestras de luma de la imagen actual y la profundidad de bits de las muestras de croma de la imagen actual se derivan de un primer conjunto de parámetros de secuencia, SPS, que esté activo para la capa actual en el momento de decodificar la imagen actual; y

55 el formato de croma de la imagen anterior, la profundidad de bits de las muestras de luma de la imagen anterior y la profundidad de bits de las muestras de croma de la imagen anterior se derivan de un segundo SPS que estaba activo para la capa actual en el momento de decodificando la imagen anterior.

60 9. El procedimiento de la reivindicación 5, en el que los valores respectivos de la profundidad de bits de las muestras de luma de la imagen actual, la profundidad de bits de las muestras de croma de la imagen actual, la profundidad de bits de las muestras de luma de la imagen anterior y la profundidad de bits de las muestras de croma de la imagen anterior se especifican como valores de profundidad de bits reales respectivos menos 8.

10. El procedimiento de la reivindicación 5, en el que el primer indicador es NoOutputOfPriorPicsFlag.

65 11. El procedimiento de la reivindicación 6, que comprende además:

- 5 obtener un ancho de la imagen actual, una altura de la imagen actual y varios almacenes de imágenes asociados con la imagen actual, en el que el ancho de la imagen actual y la altura de la imagen actual se especifican en unidades de muestras de luma, y el número de almacenes de imágenes asociados con la imagen actual indica un tamaño de DPB mínimo requerido para la decodificación de la capa actual en la secuencia de video codificada a partir de la imagen actual; y
- 10 obtener un ancho de la imagen anterior, una altura de la imagen anterior y una serie de almacenes de imágenes asociados con la imagen anterior, en el que el ancho de la imagen anterior y la altura de la imagen anterior se especifican en unidades de muestras de luma, y el número de almacenes de imágenes asociados con la imagen anterior indica un tamaño mínimo de DPB requerido para la decodificación de la capa actual en una secuencia de video codificada que termina en la imagen anterior.
12. El procedimiento de la reivindicación 11, en el que determinar el valor del primer indicador comprende:
- 15 en respuesta a la determinación de que el ancho de la imagen actual y el ancho de la imagen anterior son diferentes, la altura de la imagen actual y la altura de la imagen anterior son diferentes, o el número de almacenes de imágenes asociados con la imagen actual y el número de almacenes de imágenes asociadas con la imagen anterior es diferente: establecer el valor de la primera bandera igual a 1.
- 20 13. El procedimiento de decodificación de la reivindicación 11, que comprende además:
- decodificar un elemento de sintaxis en un flujo de bits que indica si las imágenes en la DPB se deben emitir, en el que el elemento de sintaxis está asociado con la imagen actual; y
- 25 determinar el valor del primer indicador mediante:
- en respuesta a la determinación de que el ancho de la imagen actual y el ancho de la imagen anterior son diferentes, la altura de la imagen actual y la altura de la imagen anterior son diferentes, o el número de almacenes de imágenes asociados con la imagen actual y el número de almacenes de imágenes asociadas con la imagen anterior es diferente:
- 30 establecer el valor del primer indicador igual a un valor descodificado del elemento de sintaxis.
14. El procedimiento según la reivindicación 11, en el que:
- 35 el ancho de la imagen actual, la altura de la imagen actual y el número de almacenes de imágenes asociados con la imagen actual se derivan de un primer conjunto de parámetros de secuencia, SPS, que esté activo para la capa actual en el momento de decodificar la imagen actual; y
- 40 el ancho de la imagen anterior, la altura de la imagen anterior y el número de almacenes de imágenes asociados con la imagen anterior se derivan de un segundo SPS que estaba activo para la capa actual en el momento de decodificar la imagen anterior.
- 45 15. Un medio no transitorio legible por ordenador que comprende instrucciones que, cuando son ejecutadas en un procesador que comprende hardware de ordenador, hacen que el procesador realice el procedimiento de acuerdo con una cualquiera de las reivindicaciones 5 a 14.

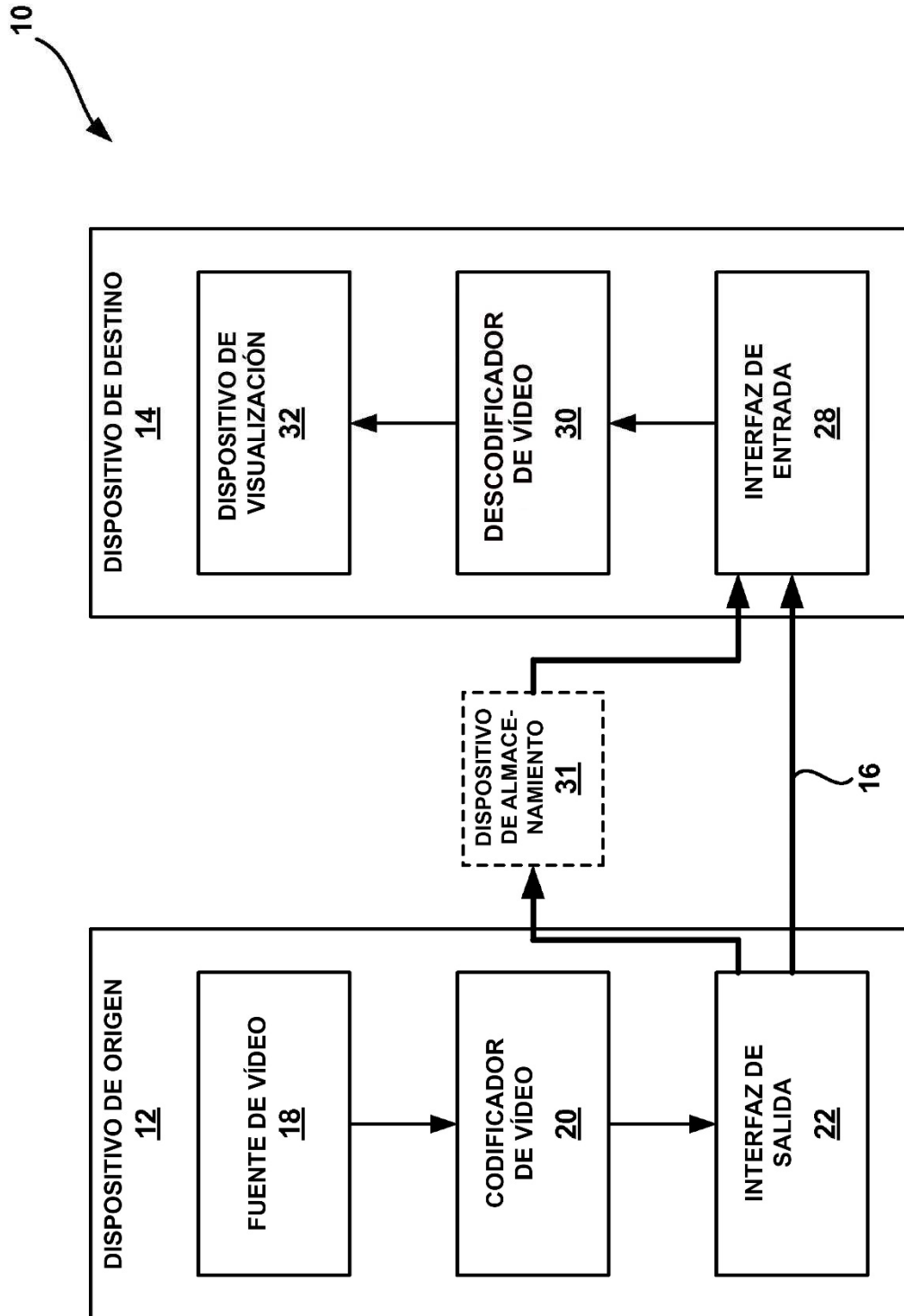


FIG. 1A

10'

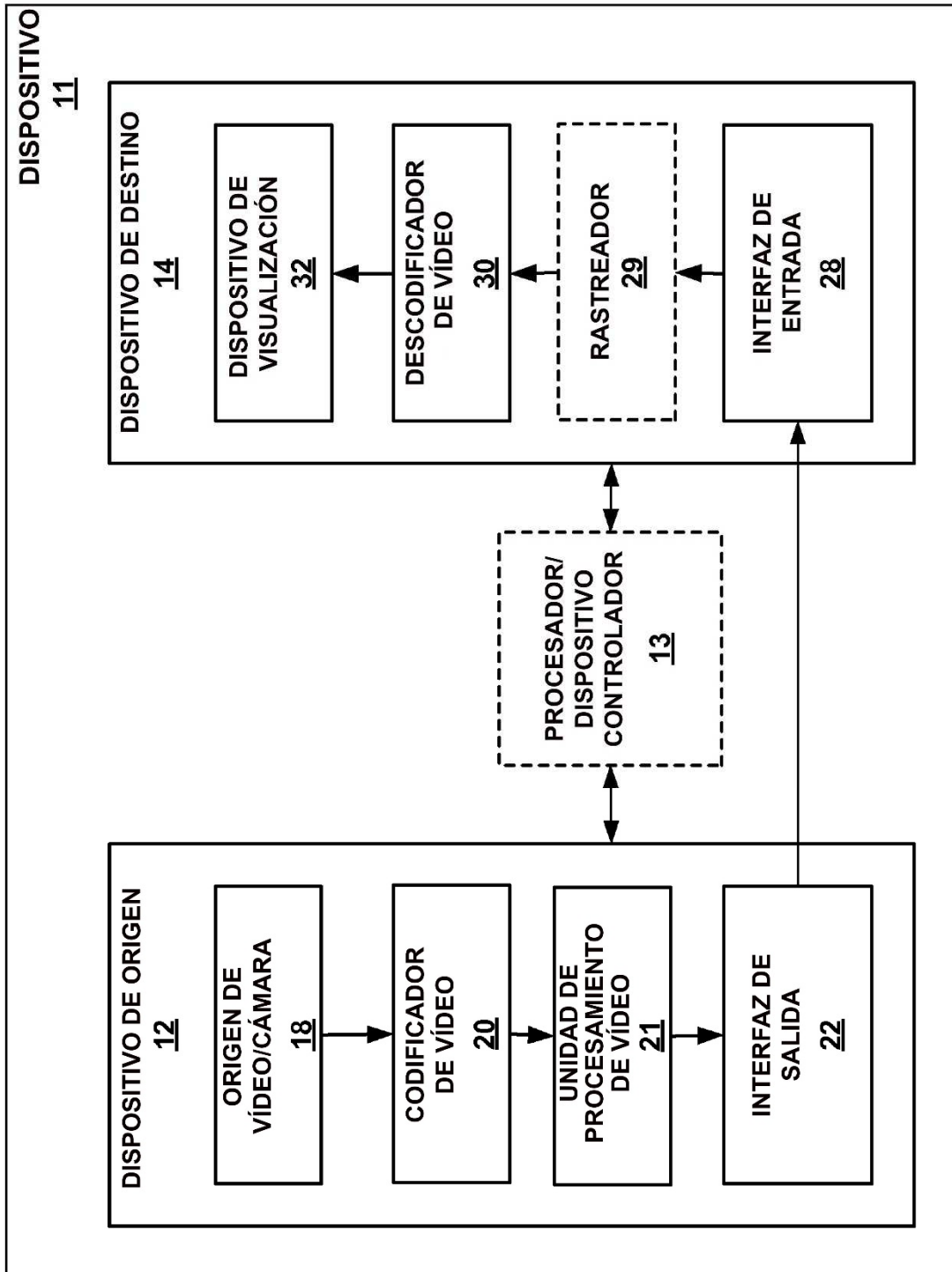


FIG. 1B

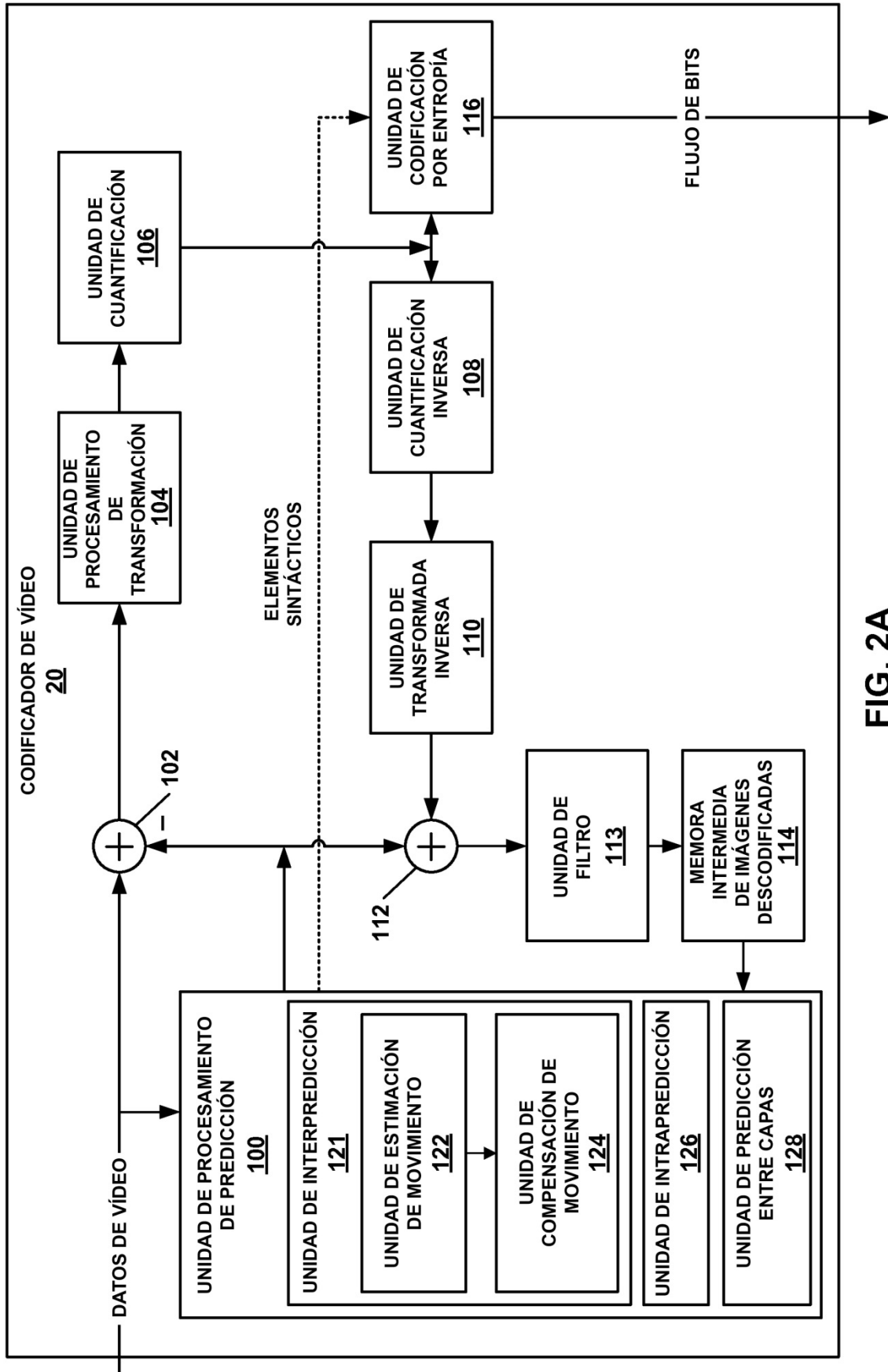


FIG. 2A

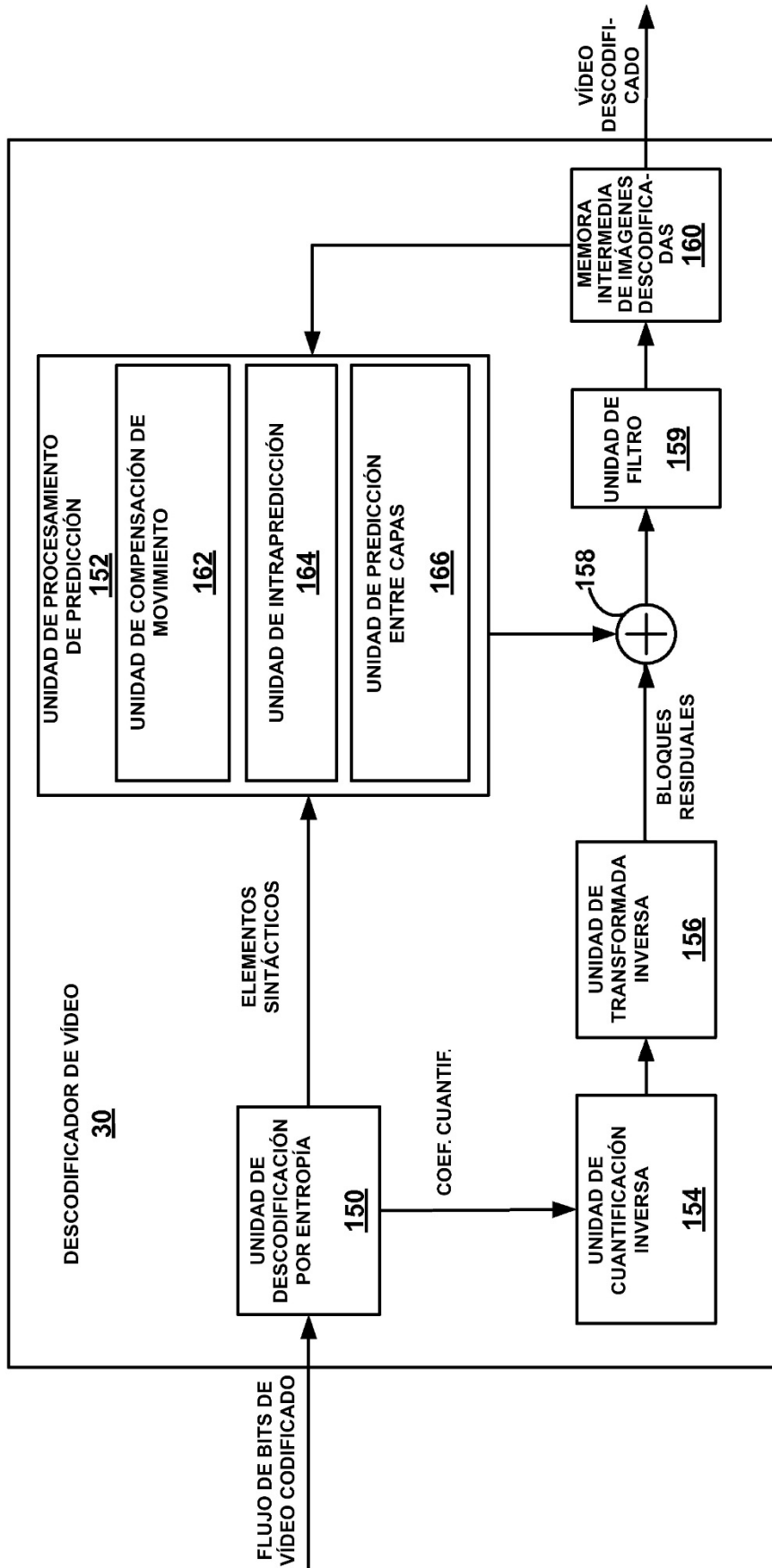


FIG. 3A

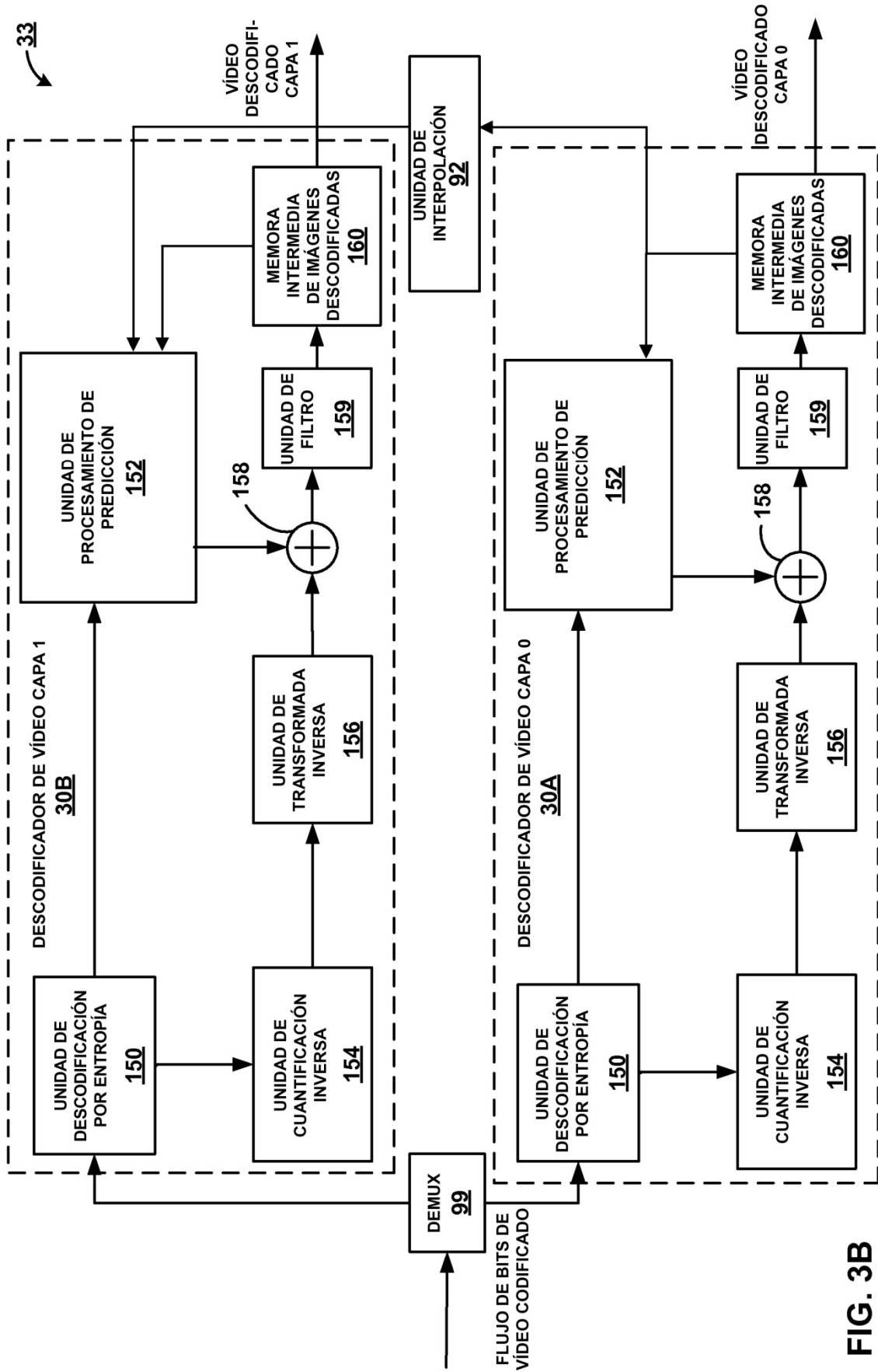


FIG. 3B

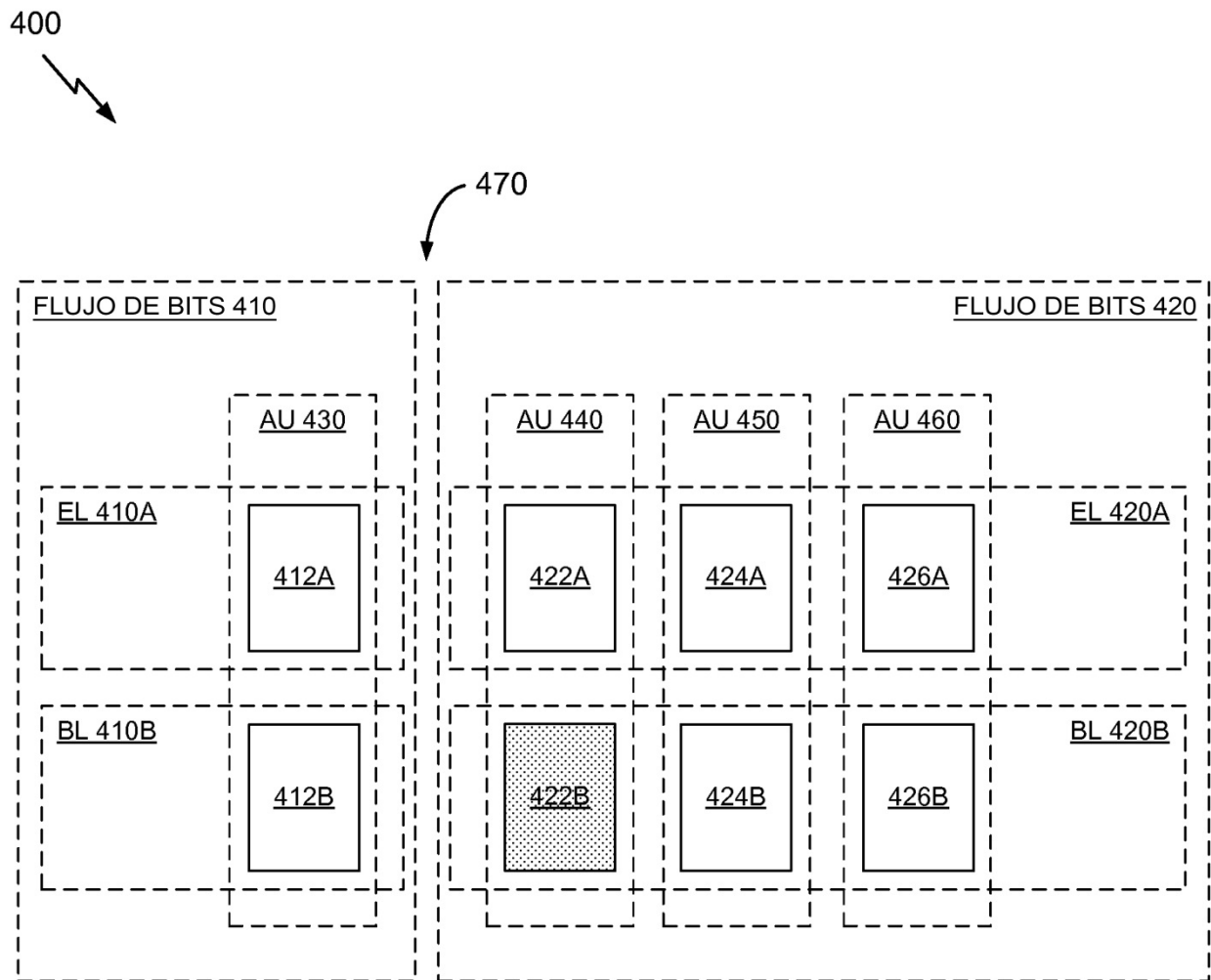


FIG. 4

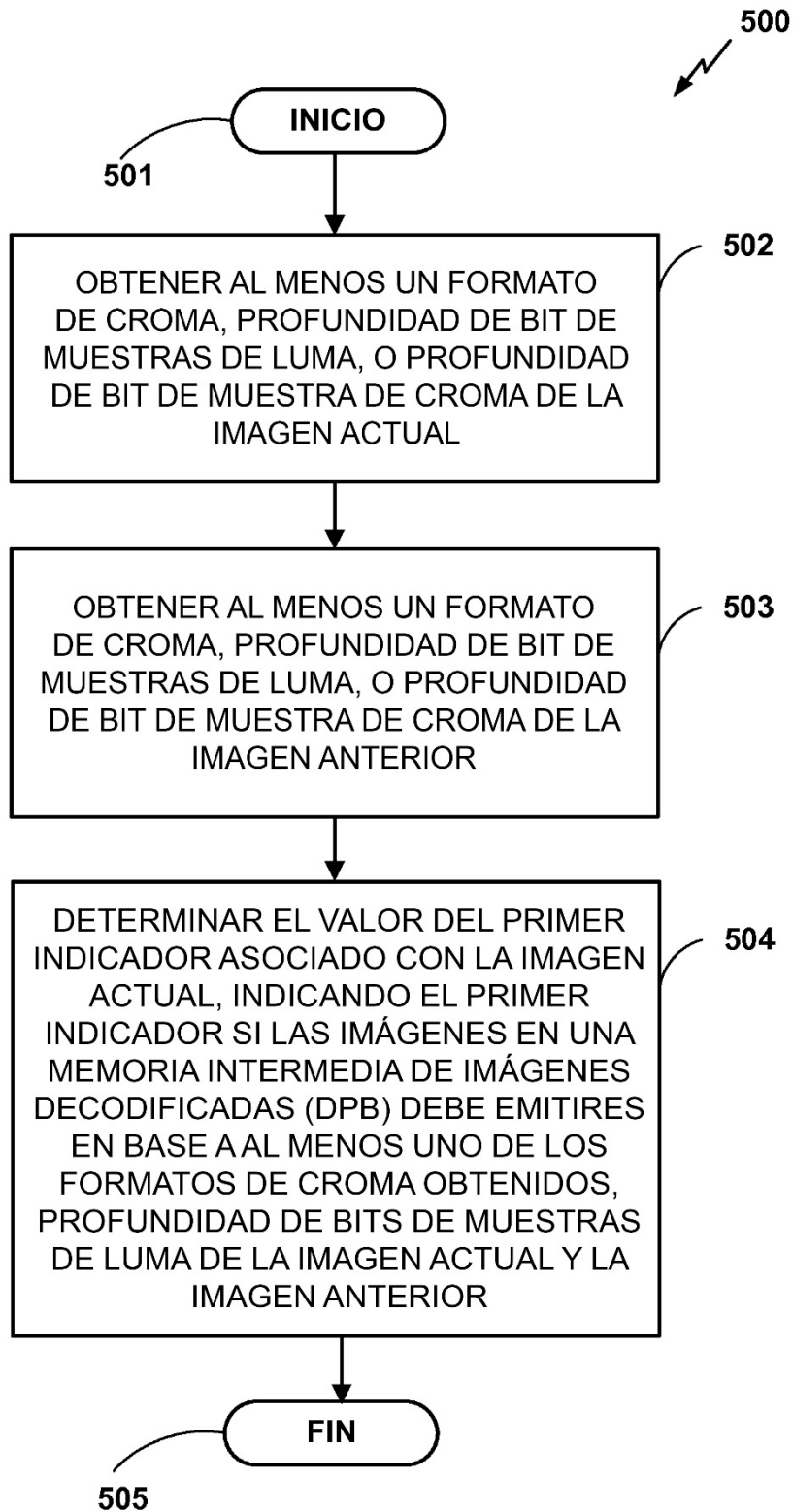


FIG. 5