

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 834 481**

51 Int. Cl.:

**H04N 19/70** (2014.01)

**H04N 19/30** (2014.01)

12

## TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **14.10.2014** **PCT/US2014/060451**

87 Fecha y número de publicación internacional: **23.04.2015** **WO15057683**

96 Fecha de presentación y número de la solicitud europea: **14.10.2014** **E 14790449 (4)**

97 Fecha y número de publicación de la concesión europea: **02.09.2020** **EP 3058728**

54 Título: **Dispositivo y procedimiento para codificación escalable de información de vídeo**

30 Prioridad:

**14.10.2013 US 201361890782 P**

**13.10.2014 US 201414513121**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:  
**17.06.2021**

73 Titular/es:

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

72 Inventor/es:

**RAMASUBRAMONIAN, ADARSH KRISHNAN;**  
**WANG, YE-KUI y**  
**CHEN, YING**

74 Agente/Representante:

**FORTEA LAGUNA, Juan José**

ES 2 834 481 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

Dispositivo y procedimiento para codificación escalable de información de vídeo

## 5 CAMPO TÉCNICO

[0001] La presente divulgación se refiere al campo de la codificación y compresión de vídeo, en particular a la codificación de vídeo escalable (SVC), la codificación de vídeo multivista (MVC) o la codificación de vídeo 3D (3DV).

## 10 ANTECEDENTES

[0002] Las capacidades de vídeo digital se pueden incorporar a una amplia gama de dispositivos, incluyendo televisores digitales, sistemas de radiodifusión directa digital, sistemas de radiodifusión inalámbrica, asistentes personales digitales (PDA), ordenadores portátiles o de escritorio, cámaras digitales, dispositivos de grabación digitales, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos de radio celulares o por satélite, dispositivos de videoconferencia y similares. Los dispositivos de vídeo digital implementan técnicas de compresión de vídeo, tales como las descritas en los estándares definidos por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, parte 10, Codificación avanzada de vídeo (AVC), el estándar de Codificación de vídeo de alta eficacia (HEVC) actualmente en desarrollo y las extensiones de dichos estándares. 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.

[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.) se puede dividir en bloques de vídeo, que también se pueden denominar 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 en un fragmento intercodificado (P o B) de una imagen pueden usar predicción espacial con respecto a muestras de referencia de bloques vecinos en la misma imagen o predicción temporal con respecto a muestras de referencia en otras imágenes de referencia. Las imágenes se pueden denominar tramas, y las imágenes de referencia se pueden denominar tramas de referencia.

[0004] La predicción espacial o temporal da como resultado un bloque predictivo para un bloque que se va a codificar. Los datos residuales representan diferencias de píxeles entre el bloque original que se va 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 los datos residuales. Para una mayor compresión, los datos residuales se pueden transformar desde el dominio de píxel en un dominio de transformada, dando como resultado coeficientes de transformada residuales, que a continuación se pueden cuantificar. Los coeficientes de transformada cuantificados, dispuestos inicialmente en una matriz bidimensional, se pueden explorar para generar un vector unidimensional de coeficientes de transformada, y se puede aplicar codificación por entropía para lograr aún más compresión. En CHOI B ET AL: "MV-HEVC/SHVC HLS: Reference picture marking and picture removal", 14 CONVENCION DE JCT-VC; 25-7-2013 - 2-8-2013; VIENA; (EQUIPO DE COLABORACIÓN CONJUNTA SOBRE CODIFICACIÓN DE VÍDEO DE ISO/IEC JTC1/SC29/WG11 E ITU-T SG.16); URL: HTTP://VVF3P3.ITU.INT/AV-ARCH/JCTVC-SITE/, no. JCTVC-N0128, 18 de julio de 2013 (2013-07-18), XP030114588, se presentan dos propuestas. En primer lugar, se proponen el marcado y la eliminación de imágenes de las imágenes de referencia remuestreadas y las imágenes descodificadas usadas para el escalado ascendente. Las imágenes de referencia remuestreadas se marcan como "no necesarias para la salida" de forma predeterminada, y se marcan como "no usadas como referencia" inmediatamente después de usarse para la predicción entre capas. Las imágenes descodificadas con la resolución espacial original se marcan como "no usadas como referencia" inmediatamente después de que se generan las imágenes de referencia remuestreadas, si esas imágenes no se usan para la interpredicción por las siguientes imágenes. En segundo lugar, se propone el proceso de eliminación de imágenes de un búfer de imágenes descodificadas (DPB). Cuando el valor de NoOutputOfPriorPicsFlag es igual a 1, se vacían todos los búferes de almacenamiento de imágenes en el DPB, excepto las imágenes que pertenecen a la misma unidad de acceso. El propósito del segundo elemento es evitar la eliminación de imágenes de referencia entre capas antes de la predicción entre capas.

## 60 BREVE EXPLICACIÓN

[0005] La invención se define en las reivindicaciones adjuntas, a las que se hace ahora referencia. La codificación de vídeo escalable (SVC) se refiere a la codificación de vídeo en la que se usan una capa base (BL), a veces denominada capa de referencia (RL), y una o más capas de mejora (EL) escalables. Para la 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 se pueden definir con respecto a una capa codificada previamente. 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 por debajo de la capa intermedia, como la capa base o cualquier capa de mejora intermedia, y al mismo tiempo servir como una RL para una o más capas de mejora por encima de la capa intermedia. De forma similar, en la ampliación multivista o 3D del estándar HEVC, puede haber múltiples vistas, y la información de una vista se puede usar para codificar (por ejemplo, codificar o decodificar) la información de otra vista (por ejemplo, estimación de movimiento, predicción del vector de movimiento y/u otras redundancias).

**[0006]** Las imágenes que se descodifican (por ejemplo, para que se puedan visualizar o usar para predecir otras imágenes) se almacenan en un búfer de imágenes descodificadas (DPB). Las imágenes que se van a enviar se pueden marcar como "necesarias para la salida", y las imágenes que se van a usar para predecir otras imágenes se pueden marcar como "usadas como referencia". Las imágenes descodificadas que no están marcadas como "necesarias para la salida" ni como "usadas como referencia" pueden estar presentes en el DPB hasta que se eliminan por el proceso de descodificación. En los descodificadores conformes con el orden de salida, el proceso de eliminación de imágenes del DPB a menudo sigue inmediatamente a la salida de imágenes que están marcadas como "necesarias para la salida". Este proceso de emisión y posterior eliminación se puede denominar "expulsión".

**[0007]** Adicionalmente, también puede haber situaciones en las que el descodificador puede eliminar las imágenes en el DPB sin salida, aunque estas imágenes puedan estar marcadas como "necesarias para la salida". Por ejemplo, para determinadas imágenes de puntos de acceso aleatorios que están en el medio de un flujo de bits, en el momento de codificar dichas imágenes, se pueden eliminar todas las imágenes en el DPB.

**[0008]** Sin embargo, en el contexto de flujos de bits multicapa, pueden surgir complicaciones si se eliminan todas las imágenes en el DPB para dichas imágenes de puntos de acceso aleatorios porque, con la posibilidad de tener imágenes IRAP no alineadas en el flujo de bits, es posible que otras imágenes en la misma unidad de acceso no sean imágenes de puntos de acceso aleatorios y, por tanto, es posible que necesiten usar algunas de las imágenes del DPB.

**[0009]** Por tanto, se desea un procedimiento de codificación mejorado para descargar el DPB cuando las imágenes de acceso aleatorio no están alineadas entre las múltiples capas.

**[0010]** Cada uno de los sistemas, procedimientos y dispositivos de la presente divulgación tiene varios aspectos innovadores, ninguno de los cuales es el único responsable de los atributos deseables divulgados en el presente documento.

**[0011]** En un aspecto, un aparato configurado para codificar (por ejemplo, codificar o decodificar) información de vídeo incluye una unidad de memoria y un procesador en comunicación con la unidad de memoria. La unidad de memoria se configura para almacenar información de vídeo asociada a una capa base (BL) y una capa de mejora (EL), teniendo la BL una imagen de BL en una primera unidad de acceso y teniendo la EL una imagen de EL en la primera unidad de acceso, en la que la imagen de BL tiene un indicador asociado a ella. El procesador se configura para determinar un valor del indicador asociado a la imagen de BL y realizar, en base al valor del indicador, uno de (1) eliminar una o más imágenes de EL en un búfer de imágenes descodificadas (DPB) sin emitir la una o más imágenes de EL antes de que se codifique la imagen de EL, o (2) abstenerse de eliminar la una o más imágenes de EL en el DPB sin emitir la una o más imágenes de EL.

**[0012]** En otro aspecto, un procedimiento de codificación de información de vídeo comprende determinar un valor de un indicador asociado a una imagen de BL en una primera unidad de acceso y realizar, en base al valor del indicador, uno de (1) eliminar una o más imágenes de EL en un búfer de imágenes descodificadas (DPB) sin emitir la una o más imágenes de EL antes de que se codifique una imagen de EL en la primera unidad de acceso, o (2) abstenerse de eliminar la una o más imágenes de EL en el DPB sin emitir la una o más imágenes de EL.

**[0013]** En otro aspecto, un medio no transitorio legible por ordenador comprende código que, cuando se ejecuta, hace que un aparato realice un proceso. El proceso incluye almacenar información de vídeo asociada con una capa base (BL) y una capa de mejora (EL), teniendo la BL una imagen de BL en una primera unidad de acceso, y teniendo la EL una imagen de EL en la primera unidad de acceso, en el que la imagen de BL tiene un indicador asociado a ella, determinar un valor del indicador asociado a la imagen de BL y realizar, en base al valor del indicador, uno de (1) eliminar una o más imágenes de EL en un búfer de imágenes descodificadas (DPB) sin emitir la una o más imágenes de EL antes de que se codifique la imagen de EL, o (2) abstenerse de eliminar la una o más imágenes de EL en el DPB sin emitir la una o más imágenes de EL.

**[0014]** En otro aspecto, un dispositivo de codificación de vídeo configurado para codificar información de vídeo comprende medios para almacenar información de vídeo asociada a una capa base (BL) y una capa de mejora (EL), teniendo la BL una imagen de BL en una primera unidad de acceso, y teniendo la EL una imagen de EL en

la primera unidad de acceso, en el que la imagen de BL tiene un indicador asociado a ella, medios para determinar un valor del indicador asociado a la imagen de BL y medios para realizar, en base al valor del indicador, uno de (1) eliminar una o más imágenes de EL en un búfer de imágenes descodificadas (DPB) sin emitir la una o más imágenes de EL antes de que se codifique la imagen de EL, o (2) abstenerse de eliminar la una o más imágenes de EL en el DPB sin emitir la una o más imágenes de EL.

## BREVE DESCRIPCIÓN DE LAS FIGURAS

### [0015]

La FIG. 1A es un diagrama de bloques que ilustra un sistema de codificación y descodificación de vídeo de ejemplo que puede usar técnicas de acuerdo con los aspectos descritos en la presente 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 de acuerdo con aspectos descritos en la presente 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 la presente 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 la presente 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 la presente 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 la presente 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.

La FIG. 6 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

[0016] Determinados modos de realización descritos en el presente documento se refieren a la predicción entre capas para la codificación de vídeo escalable en el contexto de los códecs de vídeo avanzados, tal como la HEVC (codificación de vídeo de alta eficacia). Más específicamente, la presente divulgación se refiere a sistemas y procedimientos para mejorar el rendimiento de la predicción entre capas en la ampliación de la codificación de vídeo escalable (SVC) de la HEVC.

[0017] En la siguiente descripción, se describen técnicas de H.264/AVC relacionadas con determinados modos de realización, y también se analizan el estándar HEVC y técnicas relacionadas. Aunque determinados modos de realización se describen en el presente documento en el contexto de los estándares HEVC y/o H.264, un experto en la técnica puede apreciar que los sistemas y procedimientos divulgados en el presente documento se pueden aplicar a cualquier estándar de codificación de vídeo adecuado. Por ejemplo, los modos de realización divulgados en el presente documento pueden ser aplicables a uno o más de los siguientes estándares: ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 o ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual e ITU-T H.264 (también conocido como ISO/IEC MPEG-4 AVC), incluyendo sus ampliaciones de Codificación de vídeo escalable (SVC) y Codificación de vídeo multivista (MVC).

[0018] HEVC, en general, sigue el marco de los estándares de codificación de vídeo anteriores en muchos aspectos. La unidad de predicción en HEVC es diferente de la de determinados estándares de codificación de vídeo previos (por ejemplo, un macrobloque). De hecho, el concepto de macrobloque no existe en HEVC como se entiende en determinados estándares de codificación de vídeo previos. Un macrobloque se reemplaza por una estructura jerárquica basada en un sistema de árbol cuaternario, lo cual puede proporcionar una gran flexibilidad, entre otros beneficios posibles. Por ejemplo, dentro del esquema de HEVC, se definen tres tipos de bloques, unidad de codificación (CU), unidad de predicción (PU) y unidad de transformada (TU). La CU se puede referir a la unidad básica de división de la región. La CU se puede considerar 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 se puede considerar la unidad básica de inter/intrapredicción, y una única PU puede contener múltiples particiones de forma arbitraria para codificar eficazmente patrones de imagen

irregulares. La TU se puede considerar como la unidad básica de transformada. 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 la función respectiva de la unidad, lo que puede dar como resultado una mejora de la eficacia de codificación.

**[0019]** Para propósitos de ilustración solamente, determinados 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). Se debe entender 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 determinados 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, tal como bloques (por ejemplo, CU, PU, TU, macrobloques, etc.), fragmentos, tramas, etc.

#### Estándares de codificación de vídeo

**[0020]** Una imagen digital, tal como una imagen de vídeo, una imagen de televisión, 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 típicamente es de decenas de miles. Cada píxel contiene típicamente información de luminancia y crominancia. Sin compresión, la gran cantidad de información que se debe transmitir desde un codificador de imagen a un decodificador de imagen 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 los estándares JPEG, MPEG y H.263.

**[0021]** Los estándares de codificación de vídeo incluyen ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 o ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual e ITU-T H.264 (también conocido como ISO/IEC MPEG-4 AVC), incluyendo sus ampliaciones de Codificación de vídeo escalable (SVC) y Codificación de vídeo multivista (MVC).

**[0022]** Además, existe un nuevo estándar de codificación de vídeo, a saber, la Codificación de vídeo de alta eficacia (HEVC), que está siendo desarrollado por el Equipo de colaboración conjunta en codificación de vídeo (JCTVC) 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 convención: Ginebra, Suiza, del 14 de enero de 2013 al 23 de enero de 2013. La ampliación multivista para HEVC, a saber, MV-HEVC, y la ampliación escalable para HEVC, a saber, 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 la codificación de vídeo 3D) y JCT-VC, respectivamente.

**[0023]** A continuación, en el presente documento, se describen de forma más detallada diversos aspectos de los sistemas, aparatos y procedimientos novedosos, con referencia a las figuras adjuntas. Sin embargo, la presente divulgación se puede realizar de muchas formas diferentes y no se debería interpretar que está limitada a alguna estructura o función específica presentada a lo largo de la presente divulgación. En cambio, estos aspectos se proporcionan para que esta divulgación sea exhaustiva y completa, y transmita por completo el alcance de la divulgación a los expertos en la técnica. En base a las enseñanzas del presente documento, un experto en la técnica 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. Se debe entender que cualquier aspecto divulgado en el presente documento se puede realizar mediante uno o más elementos de una reivindicación.

**[0024]** Aunque en el presente documento se describen aspectos particulares, muchas variaciones y permutaciones de estos aspectos caen dentro del alcance de la divulgación. Aunque se mencionan algunos beneficios y ventajas de los aspectos preferentes, el alcance de la divulgación no pretende limitarse a beneficios, usos u objetivos particulares. En cambio, los aspectos de la divulgación pretenden 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 las figuras son meramente ilustrativas de la divulgación, en lugar de limitativos, estando definido el alcance de la divulgación por las reivindicaciones adjuntas y equivalentes de las mismas.

[0025] Las figuras adjuntas ilustran ejemplos. Los elementos indicados por números de referencia en las figuras adjuntas corresponden a elementos indicados por 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", y así sucesivamente) no implican necesariamente que los elementos tienen un orden particular. En cambio, dichas palabras ordinales se usan simplemente para referirse a diferentes elementos de un mismo tipo o un tipo similar.

#### Sistema de codificación de vídeo

[0026] 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 la presente divulgación. Como se usa y describe en el presente documento, el término "codificador de vídeo" se refiere, en general, tanto a codificadores de vídeo como a descodificadores de vídeo. En la presente divulgación, los términos "codificación de vídeo" o "codificación" se pueden referir 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 se pueden extender 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 dispositivos intermedios (por ejemplo, dispositivos que pueden modificar, transformar y/o manipular de otro modo un flujo de bits).

[0027] Como se muestra en la **FIG. 1A**, el sistema de codificación de vídeo 10 incluye un módulo de origen 12 que genera datos de vídeo codificados que se deben descodificar posteriormente por un módulo de destino 14. En el ejemplo de la **FIG. 1A**, el módulo de origen 12 y el módulo de destino 14 están en dispositivos separados; específicamente, el módulo de origen 12 es parte de un dispositivo de origen, y el módulo de destino 14 es parte de un dispositivo de destino. Sin embargo, se observa que los módulos 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**.

[0028] Con referencia una vez más a la **FIG. 1A**, el módulo de origen 12 y el módulo de destino 14 pueden comprender cualquiera de una amplia gama de dispositivos, incluyendo 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 continua de vídeo o similares. En algunos casos, el módulo de origen 12 y el módulo de destino 14 pueden estar equipados para la comunicación inalámbrica.

[0029] El módulo de destino 14 puede recibir los datos de vídeo codificados que se deben descodificar por medio de un enlace 16. El enlace 16 puede comprender cualquier tipo de medio o dispositivo que puede desplazar los datos de vídeo codificados desde el módulo de origen 12 al módulo de destino 14. En el ejemplo de la **FIG. 1A**, el enlace 16 puede comprender un medio de comunicación para permitir que el módulo de origen 12 transmita datos de vídeo codificados directamente al módulo de destino 14 en tiempo real. Los datos de vídeo codificados se pueden modular de acuerdo con un estándar de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitirse al módulo de destino 14. El medio de comunicación puede comprender cualquier medio de comunicación inalámbrica o por cable, 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 módulo de origen 12 al módulo de destino 14.

[0030] De forma alternativa, los datos codificados se pueden emitir desde una interfaz de salida 22 hasta un dispositivo de almacenamiento 31 opcional. De forma similar, se puede acceder a los datos codificados del dispositivo de almacenamiento 31 por una interfaz de entrada 28. El dispositivo de almacenamiento 31 puede incluir cualquiera de una variedad 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 cualesquiera de almacenamiento digital adecuados 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 por el módulo de origen 12. El módulo de destino 14 puede acceder a los datos de vídeo almacenados del dispositivo de almacenamiento 31, por medio de transmisión continua o descarga. El servidor de archivos puede ser cualquier tipo de servidor que pueda almacenar datos de vídeo codificados y transmitir esos datos de vídeo codificados al módulo de destino 14. Los servidores de archivos de ejemplo incluyen un servidor web (por ejemplo, para un sitio web), un servidor FTP, dispositivos de almacenamiento conectados en red (NAS) o una unidad de disco local. El módulo de destino 14 puede acceder a los datos de vídeo codificados a través de cualquier conexión de datos estándar, incluyendo una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión Wi-Fi), una conexión cableada (por ejemplo, DSL, módem de cable, etc.) o una combinación de ambas que sea adecuada para acceder a datos de vídeo codificados almacenados en un servidor de archivos. La transmisión de datos de vídeo codificados desde el dispositivo de almacenamiento 31 puede ser una transmisión continua, una transmisión de descarga o una combinación de ambas.

**[0031]** Las técnicas de la presente divulgación no se limitan a aplicaciones o configuraciones inalámbricas. Las técnicas se pueden aplicar a la codificación de vídeo para admitir cualquiera de una variedad de aplicaciones multimedia, tales como radiodifusiones de televisión por aire (over-the-air)), transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones continuas de vídeo, por ejemplo, por medio de Internet (por ejemplo, transmisión continua dinámica adaptativa por HTTP (DASH), 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 se puede configurar 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.

**[0032]** En el ejemplo de la **FIG. 1A**, el módulo de origen 12 incluye una fuente de vídeo 18, un codificador de vídeo 20 y una interfaz de salida 22. En algunos casos, la interfaz de salida 22 puede incluir un modulador/demodulador (módem) y/o un transmisor. En el módulo de origen 12, la fuente de vídeo 18 puede incluir una fuente tal como un dispositivo de captura de vídeo, por ejemplo, una videocámara, un archivo de vídeo que contiene vídeo capturado previamente, 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 dichas fuentes. Como un ejemplo, si la fuente de vídeo 18 es una videocámara, el módulo de origen 12 y el módulo de destino 14 pueden formar los denominados "teléfonos con cámara" o "videoteléfonos", como se ilustra en el ejemplo de la **FIG. 1B**. Sin embargo, las técnicas descritas en la presente divulgación pueden ser aplicables a la codificación de vídeo en general, y se pueden aplicar a aplicaciones inalámbricas y/o por cable.

**[0033]** El vídeo capturado, precapturado o generado por ordenador se puede codificar por el codificador de vídeo 20. Los datos de vídeo codificados se pueden transmitir directamente al módulo de destino 14 por medio de la interfaz de salida 22 del módulo de origen 12. Los datos de vídeo codificados se pueden almacenar también (o de forma alternativa) en el dispositivo de almacenamiento 31 para un acceso posterior por el módulo 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 ilustrado en la **FIG. 2B**, o cualquier otro codificador de vídeo descrito en el presente documento.

**[0034]** 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 módulo de destino 14 puede recibir los datos de vídeo codificados a través del enlace 16. Los datos de vídeo codificados, comunicados a través del enlace 16, o proporcionados en el dispositivo de almacenamiento 31, pueden incluir una variedad de elementos de sintaxis generados por el codificador de vídeo 20, para su uso por un descodificador de vídeo, tal como el descodificador de vídeo 30, en la descodificación de los datos de vídeo. Dichos elementos de sintaxis se pueden incluir con los datos de vídeo codificados transmitidos en un medio de comunicación, almacenarse en un medio de almacenamiento o almacenarse 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 ilustrado en la **FIG. 3A**, el descodificador de vídeo 33 ilustrado en la **FIG. 3B**, o cualquier otro descodificador de vídeo descrito en el presente documento.

**[0035]** El dispositivo de visualización 32 puede estar integrado con, o ser externo a, el módulo de destino 14. En algunos ejemplos, el módulo 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 módulo de destino 14 puede ser un dispositivo de visualización. En general, el dispositivo de visualización 32 muestra los datos de vídeo descodificados a un usuario, y puede comprender cualquiera de una variedad de 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.

**[0036]** En aspectos relacionados, la **FIG. 1B** muestra un sistema de codificación y descodificación de vídeo 10' de ejemplo en el que los módulos de origen y de destino 12, 14 están en, o forman parte de, un dispositivo o un dispositivo de usuario 11. El dispositivo 11 puede ser un equipo telefónico de mano, tal como un teléfono "inteligente" o similar. El dispositivo 11 puede incluir un módulo controlador/procesador 13 opcional en comunicación operativa con los módulos 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 se puede implementar como una parte del codificador de vídeo 20 y/o el módulo 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 de interés a rastrear se puede segmentar por una técnica descrita en relación con uno o más aspectos de la presente divulgación. En aspectos relacionados, el rastreo se puede realizar por el dispositivo de visualización 32, solo o junto con el rastreador 29. El sistema 10' de la **FIG. 1B**, y los componentes del mismo, son similares de otro modo al sistema 10 de la **FIG. 1A** y los componentes del mismo.

**[0037]** El codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar de acuerdo con un estándar de compresión de vídeo, tal como el estándar de Codificación de vídeo de alta eficacia (HEVC), actualmente en desarrollo, y se pueden adaptar a un modelo de prueba de HEVC (HM). De forma alternativa, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar de acuerdo con otros estándares privados o industriales, tales como el estándar ITU-T H.264, denominado de forma alternativa MPEG-4, Parte 10, Codificación de avanzada de vídeo (AVC), o ampliaciones de dichos estándares. Sin embargo, las técnicas de esta divulgación no se limitan a ningún estándar de codificación particular. Otros ejemplos de estándares de compresión de vídeo incluyen MPEG-2 e ITU-T H.263.

**[0038]** Aunque no se muestra en los ejemplos de las **FIGS. 1A y FIG. 1B**, cada uno del codificador de vídeo 20 y el decodificador de vídeo 30 se pueden integrar con un codificador y decodificador 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 en flujos de datos separados. Si procede, en algunos ejemplos, las unidades MUX-DEMUX se pueden ajustar al protocolo de multiplexación ITU H.223 o a otros protocolos, tales como el protocolo de datagramas de usuario (UDP).

**[0039]** Tanto el codificador de vídeo 20 como el decodificador de vídeo 30 se pueden implementar como cualquiera de una variedad de circuitos codificadores adecuados, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), matrices de puertas 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 la presente divulgación. Cada uno del codificador de vídeo 20 y el decodificador de vídeo 30 se pueden incluir en uno o más codificadores o decodificadores, cualquiera de los cuales se puede integrar como parte de un codificador/decodificador (CÓDEC) combinado en un dispositivo respectivo.

#### **Proceso de codificación de vídeo**

**[0040]** Como se menciona anteriormente de manera breve, 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 se puede denominar 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 forma 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.

**[0041]** Para generar el flujo de bits, el codificador de vídeo 20 puede realizar operaciones de codificación en cada imagen de 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, conjuntos de parámetros de imagen, conjuntos de parámetros de adaptación y otras estructuras de sintaxis. Un conjunto de parámetros de secuencia (SPS) puede contener parámetros aplicables a cero o más secuencias de imágenes. Un conjunto de parámetros de imagen (PPS) puede contener parámetros aplicables a cero o más imágenes. Un conjunto de parámetros de adaptación (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.

**[0042]** 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 se puede denominar unidad de codificación más grande (LCU). Los bloques de árbol de HEVC pueden ser aproximadamente análogos a los macrobloques de estándares anteriores, tales como la H.264/AVC. Sin embargo, un bloque de árbol no se limita necesariamente a un tamaño particular y puede incluir una o más unidades de codificación (CU). El codificador de vídeo 20 puede usar una división en árbol cuaternario para dividir los bloques de vídeo de bloques de árbol en bloques de vídeo asociados a las CU, de ahí el nombre "bloques de árbol".

**[0043]** 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.

**[0044]** Como parte de la 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 al fragmento. Los datos codificados asociados al fragmento se pueden denominar "fragmento codificado".

**[0045]** Para generar un fragmento codificado, el codificador de vídeo 20 puede realizar operaciones de codificación en cada bloque de árbol de 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.

**[0046]** Cuando el codificador de vídeo 20 genera un fragmento codificado, el codificador de vídeo 20 puede realizar operaciones de codificación (es decir, codificar) en los bloques de árbol del fragmento de acuerdo con un orden de exploración de trama (raster). Por ejemplo, el codificador de vídeo 20 puede codificar los bloques de árbol del fragmento en un orden dirigido de izquierda a derecha en la fila más alta de los bloques de árbol del fragmento, a continuación 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.

**[0047]** Como resultado de codificar los bloques de árbol de acuerdo con el orden de exploración de trama, los bloques de árbol por encima y a la izquierda de un bloque de árbol dado pueden haberse codificado, pero los bloques de árbol por debajo y a la derecha del bloque de árbol dado aún no se han codificado. En consecuencia, el codificador de vídeo 20 puede ser capaz de acceder a la información generada por la codificación de los bloques de árbol por encima y a la izquierda del bloque de árbol dado cuando se codifica el bloque de árbol dado. 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 por debajo y a la derecha del bloque de árbol dado cuando se codifica el bloque de árbol dado.

**[0048]** Para generar un bloque de árbol codificado, el codificador de vídeo 20 puede realizar de forma recursiva una 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 se puede asociar 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, y así sucesivamente. 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 está dividido en bloques de vídeo asociados a otras CU.

**[0049]** Uno o más elementos de sintaxis del 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.

**[0050]** 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 exploración en z. En otras palabras, el codificador de vídeo 20 puede codificar una CU superior izquierda, una CU superior derecha, una CU inferior izquierda y a continuación 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 a subbloques del bloque de vídeo de la CU dividida de acuerdo con el orden de exploración en z. En otras palabras, el codificador de vídeo 20 puede codificar una CU asociada a un subbloque superior izquierdo, una CU asociada a un subbloque superior derecho, una CU asociada a un subbloque inferior izquierdo y a continuación una CU asociada a un subbloque inferior derecho, en ese orden.

**[0051]** Como resultado de la codificación de las CU de un bloque de árbol de acuerdo con un orden de exploración en z, las CU por encima, por encima y a la izquierda, por encima y a la derecha, a la izquierda y por debajo y a la izquierda de una CU dada se pueden haber codificado. Las CU por debajo y a la derecha de la CU dada todavía no se han codificado. En consecuencia, el codificador de vídeo 20 puede ser capaz de acceder a información generada por la codificación de algunas CU vecinas de la CU dada cuando se codifica la CU dada. Sin embargo, el codificador de vídeo 20 puede no ser capaz de acceder a información generada por la codificación de otras CU vecinas de la CU dada cuando se codifica la CU dada.

**[0052]** 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 se puede asociar 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.

**[0053]** Cuando el codificador de vídeo 20 usa 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 en base a unas muestras

descodificadas de la imagen asociada a la PU. Si el codificador de vídeo 20 usa intrapredicción para generar bloques de vídeo predichos de las PU de una CU, la CU es una CU intrapredicha. Cuando el codificador de vídeo 20 usa 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 en base a unas muestras descodificadas de una o más imágenes distintas a la imagen asociada a la PU. Si el codificador de vídeo 20 usa interpredicción para generar bloques de vídeo predichos de las PU de una CU, la CU es una CU interpredicha.

**[0054]** Además, cuando el codificador de vídeo 20 usa 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 se puede denominar "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.

**[0055]** 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 en base a 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.

**[0056]** Además, como parte de la realización de una operación de codificación en una CU no dividida, el codificador de vídeo 20 puede realizar una 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 a las unidades de transformada (TU) de la CU. Cada TU de una CU puede estar asociada a un bloque de vídeo residual diferente.

**[0057]** El codificador de vídeo 20 puede aplicar una o más transformadas a los bloques de vídeo residuales asociados con las TU para generar bloques de coeficientes de transformada (por ejemplo, bloques de coeficientes de transformada) asociados a las TU. Conceptualmente, un bloque de coeficientes de transformada puede ser una matriz bidimensional (2D) de coeficientes de transformada.

**[0058]** Después de generar un bloque de coeficientes de transformada, el codificador de vídeo 20 puede realizar un proceso de cuantificación en el bloque de coeficientes de transformada. 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 se puede redondear a la baja hasta un coeficiente de transformada de  $m$  bits durante la cuantificación, donde  $n$  es mayor que  $m$ .

**[0059]** El codificador de vídeo 20 puede asociar cada CU a un valor de parámetro de cuantificación (QP). El valor de QP asociado a una CU puede determinar cómo cuantifica el codificador de vídeo 20 los bloques de coeficientes de transformada asociados a la CU. El codificador de vídeo 20 puede ajustar el grado de cuantificación aplicado a los bloques de coeficientes de transformada asociados a una CU ajustando el valor de QP asociado a la CU.

**[0060]** Después de que el codificador de vídeo 20 cuantifique un bloque de coeficientes de transformada, el codificador de vídeo 20 puede generar conjuntos de elementos de sintaxis que representan los coeficientes de transformada en el bloque de coeficientes de transformada cuantificados. El codificador de vídeo 20 puede aplicar operaciones de codificación por entropía, tales como operaciones de Codificación aritmética binaria adaptable al contexto (CABAC), a algunos de estos elementos de sintaxis. 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.

**[0061]** 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 de sintaxis que contiene una indicación de un tipo de datos en la unidad NAL y bytes 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 diversas estructuras de sintaxis.

**[0062]** El descodificador 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 descodificador de vídeo 30 recibe el flujo de bits, el descodificador de vídeo 30 puede realizar una operación de análisis sintáctico en el flujo de bits. Cuando el descodificador de vídeo 30 realiza la operación de análisis sintáctico, el descodificador de vídeo 30 puede extraer elementos de sintaxis del flujo de bits. El descodificador de vídeo 30 puede reconstruir las imágenes de los datos de vídeo en base a los elementos de

sintaxis extraídos del flujo de bits. El proceso para reconstruir los datos de vídeo en base a los elementos de sintaxis puede ser, en general, recíproco al proceso realizado por el codificador de vídeo 20 para generar los elementos de sintaxis.

**[0063]** Después de que el descodificador de vídeo 30 extrae los elementos de sintaxis asociados a una CU, el descodificador de vídeo 30 puede generar bloques de vídeo predichos para las PU de la CU en base a los elementos de sintaxis. Además, el descodificador de vídeo 30 puede realizar una cuantificación inversa de unos bloques de coeficientes de transformada asociados a las TU de la CU. El descodificador de vídeo 30 puede realizar transformadas inversas en los bloques de coeficientes de transformada para reconstruir bloques de vídeo residuales asociados a 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 en base a 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 en base a los elementos de sintaxis del flujo de bits.

## **Codificador de vídeo**

**[0064]** 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 la presente divulgación. El codificador de vídeo 20 se puede configurar para procesar una sola capa de una trama de vídeo, tal como para HEVC. Además, el codificador de vídeo 20 se puede configurar para realizar cualquiera o la totalidad de las técnicas de la presente divulgación. Como ejemplo, la unidad de procesamiento de predicción 100 se puede configurar para realizar cualquiera o todas las técnicas descritas en la presente divulgación. En otro modo de realización, el codificador de vídeo 20 incluye una unidad de predicción entre capas 128 opcional que se configura para realizar cualquiera o todas las técnicas descritas en la presente divulgación. En otros modos de realización, la predicción entre capas se puede realizar por 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 se puede omitir. Sin embargo, los aspectos de la presente divulgación no se limitan a lo anterior. En algunos ejemplos, las técnicas descritas en la presente divulgación se pueden compartir entre los diversos componentes del codificador de vídeo 20. En algunos ejemplos, adicionalmente o de forma alternativa, un procesador (no mostrado) se puede configurar para realizar cualquiera o todas las técnicas descritas en la presente divulgación.

**[0065]** Con propósitos explicativos, la presente divulgación describe un codificador de vídeo 20 en el contexto de la codificación HEVC. Sin embargo, las técnicas de la presente divulgación pueden ser aplicables a otros estándares o procedimientos de codificación. El ejemplo representado en la **FIG. 2A** es para un códec de una sola capa. Sin embargo, como se describirá en mayor detalle con respecto a la **FIG. 2B**, una parte o la totalidad del codificador de vídeo 20 se puede duplicar para procesar un códec multicapa.

**[0066]** El codificador de vídeo 20 puede realizar la intracodificación y la intercodificación de los 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 dada. La intercodificación se basa en la predicción temporal para reducir o eliminar la redundancia temporal de un vídeo dentro de tramas o imágenes contiguas 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), se pueden referir a cualquiera de varios modos de codificación de base temporal.

**[0067]** 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 de residuo 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, un búfer 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 se representan en el ejemplo de la **FIG. 2A** de forma separada para propósitos de la explicación.

**[0068]** El codificador de vídeo 20 puede recibir datos de vídeo. El codificador de vídeo 20 puede recibir los datos de vídeo desde 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 la realización de 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 la realización de una operación de codificación en un fragmento, el codificador de vídeo 20 puede realizar operaciones de codificación en bloques de árbol en el fragmento.

**[0069]** Como parte de la realización de 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 se puede asociar 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.

**[0070]** 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 64x64 muestras o mayor. En la presente divulgación, "NxN" y "N por N" se pueden usar de manera intercambiable para hacer referencia a las dimensiones de muestras de un bloque de vídeo en términos de dimensiones verticales y horizontales, por ejemplo, 16x16 muestras o 16 por 16 muestras. 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 de NxN tiene, en general, N muestras en una dirección vertical y N muestras en una dirección horizontal, donde N representa un valor entero no negativo.

**[0071]** 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 a 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 al subbloque puede tener cuatro nodos secundarios, cada uno de los cuales corresponde a una CU asociada a uno de los subsubbloques.

**[0072]** Cada nodo de la estructura de datos en árbol cuaternario puede contener datos de sintaxis (por ejemplo, elementos de sintaxis) 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 de sintaxis para una CU se pueden definir 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.

**[0073]** 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.

**[0074]** 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 diversos tamaños de PU. Suponiendo que el tamaño de una CU en 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 interpolación en tamaños de PU simétricos de  $2N \times 2N$ ,  $2N \times N$ ,  $N \times N$ ,  $N \times 2N$ ,  $2N \times nU$ ,  $nL \times 2N$ ,  $nR \times 2N$  o similares. El codificador de vídeo 20 y el descodificador de vídeo 30 también pueden admitir la división asimétrica 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.

**[0075]** La unidad de interpolación 121 puede realizar la interpolación en cada PU de la CU. La interpolación puede proporcionar compresión temporal. Para realizar la interpolación 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 en base a la información de movimiento y las muestras descodificadas de imágenes distintas a la imagen asociada a la CU (por ejemplo, imágenes de referencia). En la presente divulgación, un bloque de vídeo predicho generado por la unidad de compensación de movimiento 124 se puede denominar un bloque de vídeo interpolado.

**[0076]** 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 interpolación en la PU.

**[0077]** Si la PU está en un fragmento P, la imagen que contiene la PU se asocia a una lista de imágenes de referencia denominada "lista 0". Cada una de las imágenes de referencia de la lista 0 contiene muestras que se pueden usar para la interpolación 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.

**[0078]** 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 diversos 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 se pueden interpolar 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 en base al bloque de referencia identificado por la información de movimiento de la PU.

**[0079]** 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.

**[0080]** 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 dirección de predicción y el vector de movimiento como la información de movimiento de la PU. El indicador de 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 en base al bloque de referencia indicado por la información de movimiento de la PU.

**[0081]** 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 en base a los bloques de referencia indicados por la información de movimiento de la PU.

**[0082]** 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 cambio, 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 vecina. En este ejemplo, la unidad de estimación de movimiento 122 puede indicar, en una estructura de sintaxis asociada a la PU, un valor que indica al descodificador de vídeo 30 que la PU tiene la misma información de movimiento que la PU vecina. En otro ejemplo, la unidad de estimación de movimiento 122 puede identificar, en una estructura de sintaxis asociada a la PU, una PU vecina 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 vecina indicada. El descodificador de vídeo 30 puede usar el vector de movimiento de la PU vecina indicada y la diferencia de vectores de movimiento para determinar el vector de movimiento de la PU. Con 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.

**[0083]** Como se analiza en mayor detalle a continuación con referencia a las **FIGS. 5 y 6**, la unidad de

procesamiento de predicción 100 se puede configurar para codificar (por ejemplo, codificar o decodificar) la PU (o cualquier otro bloque o unidad de vídeo de capa de referencia y/o de capa de mejora) realizando los procedimientos ilustrados en las **FIGS. 5 y 6**. Por ejemplo, la unidad de interpredicción 121 (por ejemplo, por medio 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 se pueden configurar para realizar los procedimientos ilustrados en las **FIGS. 5 y 6**, ya sea juntas o por separado.

**[0084]** 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 en base a muestras decodificadas de otras PU en la misma imagen. Los datos de predicción para la PU pueden incluir un bloque de vídeo predicho y diversos elementos de sintaxis. La unidad de intrapredicción 126 puede realizar la intrapredicción en PU en fragmentos I, fragmentos P y fragmentos B.

**[0085]** 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 usa 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 vecinas a través del bloque de vídeo de la PU en una dirección y/o gradiente asociados con el modo de intrapredicción. Las PU vecinas pueden estar por encima, por encima y a la derecha, por encima 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 diversos números de modos de intrapredicción, por ejemplo, 33 modos de intrapredicción direccional, dependiendo del tamaño de la PU.

**[0086]** La unidad de procesamiento de predicción 100 puede seleccionar los datos de predicción para una PU de 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 en base a las métricas de velocidad/distorsión de los conjuntos de datos de predicción.

**[0087]** 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 usó 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 vecina. En otras palabras, el modo de intrapredicción de la PU vecina puede ser el modo más probable para la PU actual. De este modo, la unidad de procesamiento de predicción 100 puede generar un elemento de sintaxis para indicar que el modo de intrapredicción seleccionado es el mismo que el modo de intrapredicción de la PU vecina.

**[0088]** Como se analiza anteriormente, el codificador de vídeo 20 puede incluir la unidad de predicción entre capas 128. La unidad de predicción entre capas 128 se configura 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 intrapredicción entre capas, la predicción de movimiento entre capas y la predicción de residuo entre capas. La intrapredicción entre capas usa 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 de residuo 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.

**[0089]** Después de que la unidad de procesamiento de predicción 100 selecciona los datos de predicción para las PU de una CU, la unidad de generación de residuo 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 al 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.

**[0090]** 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 estar basados, 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.

**[0091]** La unidad de procesamiento de transformada 104 puede generar uno o más bloques de coeficientes de transformada para cada TU de una CU aplicando una o más transformadas a un bloque de vídeo residual asociado a la TU. Cada uno de los bloques de coeficientes de transformada puede ser una matriz 2D de coeficientes de transformada. La unidad de procesamiento de transformada 104 puede aplicar diversas transformadas al bloque de vídeo residual asociado a una TU. Por ejemplo, la unidad de procesamiento de transformada 104 puede aplicar una transformada de coseno discreta (DCT), una transformada direccional o una transformada conceptualmente similar al bloque de vídeo residual asociado a una TU.

**[0092]** Después de que la unidad de procesamiento de transformada 104 genera un bloque de coeficientes de transformada asociado con una TU, la unidad de cuantificación 106 puede cuantificar los coeficientes de transformada en el bloque de coeficientes de transformada. La unidad de cuantificación 106 puede cuantificar un bloque de coeficientes de transformada asociado a una TU de una CU en base a un valor de QP asociado a la CU.

**[0093]** El codificador de vídeo 20 puede asociar un valor de QP a una CU de diversas maneras. Por ejemplo, el codificador de vídeo 20 puede realizar un análisis de velocidad-distorsión en un bloque de árbol asociado a 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 múltiples veces en el bloque de árbol. El codificador de vídeo 20 puede asociar diferentes valores de QP a 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 de QP dado está asociado a la CU cuando el valor de QP dado está asociado a la CU en una representación codificada del bloque de árbol que tiene una velocidad de bits y una métrica de distorsión más bajas.

**[0094]** La unidad de cuantificación inversa 108 y la unidad de transformada inversa 110 pueden aplicar la cuantificación inversa y las transformadas inversas al bloque de coeficientes de transformada, respectivamente, para reconstruir un bloque de vídeo residual a partir del bloque de coeficientes de transformada. 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 a una TU. Reconstruyendo 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.

**[0095]** Después de que la unidad de reconstrucción 112 reconstruye el bloque de vídeo de una CU, la unidad de filtro 113 puede realizar una operación de eliminación de efecto pixelado para reducir los artefactos de efecto pixelado en el bloque de vídeo asociado a la CU. Después de realizar la una o más operaciones de eliminación de efecto pixelado, la unidad de filtro 113 puede almacenar el bloque de vídeo reconstruido de la CU en el búfer 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 el búfer de imágenes descodificadas 114 para realizar la intrapredicción en otras PU en la misma imagen que la CU.

**[0096]** La unidad de codificación por entropía 116 puede recibir datos desde otros componentes funcionales del codificador de vídeo 20. Por ejemplo, la unidad de codificación por entropía 116 puede recibir bloques de coeficientes de transformada de la unidad de cuantificación 106 y puede recibir elementos de sintaxis 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 de 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.

**[0097]** 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 bins particulares tengan unos valores particulares. En el contexto de CABAC, el término "bin" se usa para hacer referencia a un bit de una versión binarizada de un elemento de sintaxis.

Codificador de vídeo multicapa

[0098] La **FIG. 2B** es un diagrama de bloques que ilustra un ejemplo de un codificador de vídeo multicapa 23 que puede implementar técnicas de acuerdo con aspectos descritos en la presente divulgación. El codificador de vídeo 23 se puede configurar para procesar tramas de vídeo multicapa, tal como para SHVC y codificación multivista. Además, el codificador de vídeo 23 se puede configurar para realizar cualquiera o todas las técnicas de la presente divulgación.

[0099] El codificador de vídeo 23 incluye un codificador de vídeo 20A y un codificador de vídeo 20B, cada uno de los cuales se puede configurar 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 el reuso 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 se puede procesar o codificar 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 de codificador de vídeo pueden estar inactivas al procesar algunas unidades de acceso.

[0100] 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 una interpolación (upsampling) 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 una interpolación 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 una interpolación 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 se configura para realizar una interpolación 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 exploración de trama. Aunque se describe principalmente como una interpolación de una capa base o de una capa inferior en una unidad de acceso, en algunos casos, la unidad de remuestreo 90 puede realizar un diezmado (downsampling) de una capa. Por ejemplo, si durante la transmisión continua de un vídeo se reduce el ancho de banda, una trama se puede diezmar en lugar de interpolar.

[0101] La unidad de remuestreo 90 se puede configurar para recibir una imagen o trama (o información de imagen asociada con la imagen) desde el búfer de imágenes descodificadas 114 del codificador de capa inferior (por ejemplo, el codificador de vídeo 20A) y para realizar una interpolación de la imagen (o de la información de imagen recibida). Esta imagen interpolada se puede proporcionar a continuación 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**.

[0102] En algunos casos, la unidad de remuestreo 90 se puede omitir o ignorar. En dichos casos, la imagen del búfer 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 del búfer de imágenes descodificadas 114 del codificador de vídeo 20A son del mismo tamaño o resolución, la imagen de referencia se puede proporcionar al codificador de vídeo 20B sin remuestreo.

[0103] En algunos modos de realización, el codificador de vídeo 23 realiza un diezmado de los datos de vídeo que se deben proporcionar al codificador de capa inferior usando la unidad de diezmado 94 antes de proporcionar los datos de vídeo al codificador de vídeo 20A. De forma alternativa, la unidad de diezmado 94 puede ser una unidad de remuestreo 90 capaz de realizar una interpolación o un diezmado de los datos de vídeo. Aún en otros modos de realización, la unidad de diezmado 94 se puede omitir.

[0104] 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 dado. 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 se pueden alternar 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 se puede crear 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, se pueden enviar 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 estar preprogramado. 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 módulo de origen 12. La señal de control se puede generar 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**

**[0105]** 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 la presente divulgación. El descodificador de vídeo 30 se puede configurar para procesar una sola capa de una trama de vídeo, tal como para HEVC. Además, el descodificador de vídeo 30 se puede configurar para realizar una cualquiera o la totalidad de las técnicas de la presente divulgación. Como ejemplo, la unidad de compensación de movimiento 162 y/o la unidad de intrapredicción 164 se pueden configurar para realizar una cualquiera o la totalidad de las técnicas descritas en la presente divulgación. En un modo de realización, el descodificador de vídeo 30 puede incluir opcionalmente una unidad de predicción entre capas 166 que se configura para realizar una cualquiera o la totalidad de las técnicas descritas en la presente divulgación. En otros modos de realización, la predicción entre capas se puede realizar por 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 se puede omitir. Sin embargo, los aspectos de la presente divulgación no se limitan a lo anterior. En algunos ejemplos, las técnicas descritas en la presente divulgación se pueden compartir entre los diversos componentes del descodificador de vídeo 30. En algunos ejemplos, adicionalmente o de forma alternativa, un procesador (no mostrado) se puede configurar para realizar cualquiera o todas las técnicas descritas en la presente divulgación.

**[0106]** Con propósitos explicativos, la presente divulgación describe un descodificador de vídeo 30 en el contexto de la codificación HEVC. Sin embargo, las técnicas de la presente divulgación pueden ser aplicables a otros estándares o procedimientos de codificación. El ejemplo representado en la **FIG. 3A** es para un códec de una sola capa. Sin embargo, como se describe en mayor detalle con respecto a la **FIG. 3B**, una parte o la totalidad del descodificador de vídeo 30 se puede duplicar para el procesamiento de un códec multicapa.

**[0107]** En el ejemplo de la **FIG. 3A**, el descodificador de vídeo 30 incluye una pluralidad de componentes funcionales. Los componentes funcionales del descodificador 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 un búfer 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 descodificador 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 descodificador de vídeo 30 puede incluir más, menos o diferentes componentes funcionales.

**[0108]** El descodificador 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 de sintaxis. Cuando el descodificador 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 de sintaxis 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 de sintaxis codificados por entropía en el flujo de bits. La unidad de procesamiento de predicción 152, la unidad de cuantificación inversa 154, la unidad de transformada 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 en base a los elementos de sintaxis extraídos del flujo de bits.

**[0109]** Como se analiza 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 SEI NAL, y así sucesivamente. 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 SEI NAL, y así sucesivamente.

**[0110]** 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 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 de sintaxis pertenecientes a un fragmento. Los elementos de sintaxis de la cabecera de fragmento pueden incluir un elemento de sintaxis 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 de sintaxis de la cabecera del fragmento codificado para recuperar la cabecera del fragmento.

**[0111]** 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 de sintaxis de las CU codificadas en los datos del fragmento. Los elementos de sintaxis extraídos pueden incluir elementos de sintaxis asociados a bloques de coeficientes de transformada. La unidad de descodificación por entropía 150 puede entonces realizar operaciones de descodificación de CABAC en algunos de los elementos de sintaxis.

**[0112]** Después de que la unidad de descodificación por entropía 150 realiza 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. Al realizar la operación de reconstrucción para cada TU de la CU, el descodificador de vídeo 30 puede reconstruir un bloque de vídeo residual asociado a la CU.

**[0113]** 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 transformada asociado a la TU. La unidad de cuantificación inversa 154 puede cuantificar de forma inversa el bloque de coeficientes de transformada de una manera similar a los procesos de cuantificación inversa propuestos para HEVC o definidos por el estándar 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 transformada para determinar un grado de cuantificación y, del mismo modo, un grado de cuantificación inversa para que la unidad de cuantificación inversa 154 lo aplique.

**[0114]** Después de que la unidad de cuantificación inversa 154 cuantifica de forma inversa un bloque de coeficientes de transformada, la unidad de transformada inversa 156 puede generar un bloque de vídeo residual para la TU asociada al bloque de coeficientes de transformada. La unidad de transformada inversa 156 puede aplicar una transformada inversa al bloque de coeficientes de transformada para generar el bloque de vídeo residual para la TU. Por ejemplo, la unidad de transformada inversa 156 puede aplicar una DCT inversa, una transformada de número entero inversa, una transformada de Karhunen-Loeve (KLT) inversa, una transformada de rotación inversa, una transformada direccional inversa u otra transformada inversa al bloque de coeficientes de transformada. En algunos ejemplos, la unidad de transformada inversa 156 puede determinar una transformada inversa que se debe aplicar al bloque de coeficientes de transformada en base a la señalización del codificador de vídeo 20. En dichos ejemplos, la unidad de transformada inversa 156 puede determinar la transformada inversa en base a una transformada señalizada en el nodo raíz de un árbol cuaternario para un bloque de árbol asociado al bloque de coeficientes de transformada. En otros ejemplos, la unidad de transformada inversa 156 puede inferir la transformada 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 transformada inversa 156 puede aplicar una transformada inversa en cascada.

**[0115]** En algunos ejemplos, la unidad de compensación de movimiento 162 puede refinar el bloque de vídeo predicho de una PU mediante interpolación en base a filtros de interpolación. Los identificadores para los filtros de interpolación que se deben usar para la compensación de movimiento con una precisión de submuestra se pueden incluir en los elementos de sintaxis. 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 generar el bloque de vídeo predicho.

**[0116]** Como se analiza en mayor detalle más adelante con referencia a las **FIGS. 5 y 6**, la unidad de procesamiento de predicción 152 puede codificar (por ejemplo, codificar o descodificar) la PU (o cualquier otra capa, bloque o unidad de vídeo de capa de referencia y/o capa de mejora) realizando los procedimientos ilustrados en las **FIGS. 5 y 6**. Por ejemplo, la unidad de compensación de movimiento 162, la unidad de intrapredicción 164 o la unidad de predicción entre capas 166 pueden estar configuradas para realizar los procedimientos ilustrados en las **FIGS. 5 y 6**, ya sea juntas o por separado.

[0117] 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 en base a los elementos de sintaxis del flujo de bits. El flujo de bits puede incluir elementos de sintaxis que la unidad de intrapredicción 164 puede usar para determinar el modo de intrapredicción de la PU.

[0118] En algunos casos, los elementos de sintaxis pueden indicar que la unidad de intrapredicción 164 debe 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 vecina. En otras palabras, el modo de intrapredicción de la PU vecina 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 de sintaxis que indica que el modo de intrapredicción de la PU es el mismo que el modo de intrapredicción de la PU vecina. La unidad de intrapredicción 164 puede entonces usar el modo de intrapredicción para generar datos de predicción (por ejemplo, muestras predichas) para la PU en base a los bloques de vídeo de las PU espacialmente vecinas.

[0119] Como se analiza anteriormente, el descodificador de vídeo 30 también puede incluir una unidad de predicción entre capas 166. La unidad de predicción entre capas 166 se configura 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 intrapredicción entre capas, la predicción de movimiento entre capas y la predicción de residuo entre capas. La intrapredicción entre capas usa 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 de residuo 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.

[0120] 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. Por tanto, el descodificador de vídeo 30 puede generar un bloque de vídeo predicho y un bloque de vídeo residual en base a unos elementos de sintaxis del flujo de bits y puede generar un bloque de vídeo en base al bloque de vídeo predicho y el bloque de vídeo residual.

[0121] Después de que la unidad de reconstrucción 158 reconstruye el bloque de vídeo de la CU, la unidad de filtro 159 puede realizar una operación de eliminación de efecto pixelado para reducir los artefactos de efecto pixelado asociados a la CU. Después de que la unidad de filtro 159 realiza una operación de eliminación de efecto pixelado para reducir los artefactos de efecto pixelado asociados a la CU, el descodificador de vídeo 30 puede almacenar el bloque de vídeo de la CU en el búfer de imágenes descodificadas 160. El búfer 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, en base a los bloques de vídeo del búfer de imágenes descodificadas 160, operaciones de intrapredicción o de interpredicción en las PU de otras CU.

### **Descodificador multicapa**

[0122] La FIG. 3B es un diagrama de bloques que ilustra un ejemplo de descodificador de vídeo multicapa 33 que puede implementar técnicas de acuerdo con aspectos descritos en la presente divulgación. El descodificador de vídeo 33 se puede configurar para procesar tramas de vídeo multicapa, tal como para SHVC y la codificación multivista. Además, el descodificador de vídeo 33 se puede configurar para realizar cualquiera o todas las técnicas de la presente divulgación.

[0123] El descodificador de vídeo 33 incluye un descodificador de vídeo 30A y un descodificador de vídeo 30B, cada uno de los cuales se puede configurar 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 el reuso 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 se puede procesar o descodificar por 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.

[0124] Además de los descodificadores de vídeo 30A y 30B, el descodificador de vídeo 33 puede incluir una unidad de interpolación 92. En algunos modos de realización, la unidad de interpolación 92 puede interpolar una capa base de una trama de vídeo recibida para crear una capa mejorada que se añadirá a la lista de imágenes de referencia para la trama o la unidad de acceso. Esta capa mejorada se puede almacenar en el búfer 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 interpolación 92 se configura para interpolar 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 exploración 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.

[0125] La unidad de interpolación 92 puede estar configurada para recibir una imagen o trama (o información de imagen asociada a la imagen) desde el búfer de imágenes descodificadas 160 del descodificador de capa inferior (por ejemplo, el descodificador de vídeo 30A) y para interpolar la imagen (o la información de imagen recibida). Esta imagen interpolada se puede proporcionar a continuación 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**.

[0126] En algunos casos, la unidad de interpolación 92 se puede omitir o ignorar. En dichos casos, la imagen del búfer 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 del búfer de imágenes descodificadas 160 del descodificador de vídeo 30A son del mismo tamaño o resolución, la imagen de referencia se puede proporcionar al descodificador de vídeo 30B sin interpolación. Además, en algunos modos de realización, la unidad de interpolación 92 puede ser una unidad de remuestreo 90 configurada para realizar la interpolación o el diezmado de la imagen de referencia recibida del búfer de imágenes descodificadas 160 del descodificador de vídeo 30A.

[0127] Como se ilustra en la **FIG. 3B**, el descodificador de vídeo 33 puede incluir además un desmultiplexor 99 o demux. El demux 99 puede dividir un flujo de bits de vídeo codificado en múltiples flujos de bits, proporcionándose cada flujo de bits enviado por el demux 99 a un descodificador de vídeo diferente 30A y 30B. Los múltiples flujos de bits se pueden crear recibiendo 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 dado. Mientras que en algunos casos los bits del flujo de bits recibidos en el demux 99 se pueden alternar 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 caso 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 se puede dividir por una proporción de bloques diferente a 1:1 a cada uno de los descodificadores de vídeo 30A y 30B. Por ejemplo, se pueden proporcionar 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 se puede preprogramar. 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 se puede generar 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)**

[0128] Algunos esquemas de codificación de vídeo pueden proporcionar diversos puntos de acceso aleatorios en todo el flujo de bits de modo que el flujo de bits se puede descodificar comenzando desde cualquiera de esos puntos de acceso aleatorio sin necesidad de descodificar ninguna imagen que preceda esos puntos de acceso aleatorio en el flujo de bits. En dichos esquemas de codificación de vídeo, todas las imágenes que siguen a un punto de acceso aleatorio en 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 continua, operaciones de búsqueda, conmutación de canales, etc.

[0129] En algunos esquemas de codificación, dichos puntos de acceso aleatorio se pueden proporcionar por

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 ("layerA") contenida en una unidad de acceso ("auA") puede proporcionar acceso aleatorio específico de capa de modo que para cada capa de referencia ("layerB") de la layerA (por ejemplo, siendo una capa de referencia una capa que se usa para predecir layerA) que tiene un punto de acceso aleatorio contenido en una unidad de acceso ("auB") que está en layerB y precede a auA en orden de descodificación (o a un punto de acceso aleatorio contenido en auA), las imágenes en layerA que siguen a auB en orden de salida (incluyendo las imágenes localizadas en auB) se pueden descodificar correctamente sin necesidad de descodificar ninguna imagen en layerA que preceda a auB.

**[0130]** Las imágenes IRAP se pueden codificar usando la intrapredicción (por ejemplo, codificarse sin hacer referencia a otras imágenes) y/o la predicción entre capas, y pueden incluir, por ejemplo, imágenes de actualización instantánea del descodificador (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 orden de descodificación no se usan para la predicción por las imágenes que siguen a la imagen IDR. 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 orden de descodificación para la predicción. Dichas imágenes que siguen a la imagen CRA en orden de descodificación, pero usan imágenes que preceden a la imagen CRA en orden de descodificación, se pueden denominar imágenes principales omitidas de acceso aleatorio (RASL). Otro tipo de imagen que puede seguir a una imagen de IRAP en orden de descodificación y precederla en 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 de IRAP en orden de descodificación. Las imágenes RASL se pueden descartar 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 orden de descodificación). Una unidad de acceso (por ejemplo, un grupo de imágenes que consisten en todas las imágenes codificadas asociadas al mismo tiempo de salida en las múltiples capas) que contiene una imagen de capa base (por ejemplo, que tiene una ID de capa de 0) que es una imagen IRAP se puede denominar una unidad de acceso IRAP.

#### **Alineación de imágenes IRAP de capas cruzadas**

**[0131]** 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 dichas 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 eficacia de codificación.

**[0132]** 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 dichos esquemas de codificación pueden hacer que los valores de POC se restablezcan (por ejemplo, en cero o en algún valor señalado en el flujo de bits) siempre que determinados tipos de imágenes aparezcan en el flujo de bits. Por ejemplo, los valores de POC de determinadas imágenes IRAP se pueden restablecer, lo que hace que los valores de POC de otras imágenes que preceden a esas imágenes IRAP en 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 de 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 de 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 tanto, en este ejemplo, el proceso de derivación para derivar los valores de POC de picC y picD se puede modificar para producir valores de POC que sean consecuentes con la definición de valores de POC y unidades de acceso.

#### **Imagen de inicialización de capa (LIP)**

**[0133]** 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 indicador NoRasIOutputFlag (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) establecido en 1 o una imagen que está contenida en una unidad de acceso IRAP inicial, que es una unidad de acceso IRAP en la que la imagen de 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 NoRasIOutputFlag establecido

en 1.

**[0134]** En algunos modos de realización, se puede activar un SPS en cada imagen LIP. Por ejemplo, cada imagen IRAP que tiene un indicador NoRasOutputFlag 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, especificando diferentes resoluciones de imagen, etc.) del SPS que se activó 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 capa base en la unidad de acceso IRAP inicial es una imagen IDR con un indicador NoCilasOutputFlag (por ejemplo, un indicador que indica que las imágenes de omisión de acceso aleatorio de capas cruzadas no se deben emitir si se establece en 1 e indica que las imágenes de omisión de acceso aleatorio de capas cruzadas se deben emitir si se establece en 0) establecido en 0, no se debe permitir que la imagen LIP active un nuevo SPS. Si se activa un nuevo SPS en dicha imagen LIP en dicho caso, en particular cuando el contenido de la SPS RBSP del nuevo SPS es diferente al del SPS que estaba activo previamente antes de la unidad de acceso IRAP inicial, podría haber problemas en diferentes resoluciones de imagen y recuperación frente a errores. Por ejemplo, el nuevo SPS puede actualizar la resolución y usar la predicción temporal para referirse a imágenes de diferentes tamaños.

### **Expulsión y descarga de imágenes**

**[0135]** Las imágenes que se descodifican (por ejemplo, para que se puedan visualizar o usar para predecir otras imágenes) se almacenan en un búfer de imágenes descodificadas (DPB). Las imágenes que se van a emitir se pueden marcar como "necesarias para la salida", y las imágenes que se van a usar para predecir otras imágenes se pueden marcar como "usadas como referencia". Las imágenes descodificadas que no están marcadas como "necesarias para la salida" ni como "usadas como referencia" (por ejemplo, imágenes que se marcaron inicialmente como "usadas como referencia" o "necesarias para la salida" pero posteriormente se marcaron como "no usadas como referencia" o "no necesarias para la salida") pueden estar presentes en el DPB hasta que se eliminan por el proceso de descodificación. En los descodificadores conformes con el orden de salida, el proceso de eliminación de imágenes del DPB a menudo sigue inmediatamente a la salida de imágenes que están marcadas como "necesarias para la salida". Este proceso de emisión y posterior eliminación se puede denominar "expulsión".

**[0136]** También hay situaciones donde el descodificador puede eliminar las imágenes en el DPB sin emitirlas, aunque estas imágenes pueden estar marcadas como "necesarias para la salida". Para facilitar la descripción en el presente documento, las imágenes descodificadas que están presentes en el DPB en el momento de descodificar una imagen IRAP (independientemente de si las imágenes descodificadas están marcadas como "necesarias para la salida" o "usadas como referencia") se denominan "imágenes del DPB con retardo" asociadas con la imagen IRAP o "imágenes del DPB con retardo asociadas" de la imagen IRAP. Algunos ejemplos de dichas situaciones, en el contexto de HEVC, se describen a continuación.

**[0137]** 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 es la primera imagen en el flujo de bits), las imágenes del DPB con retardo asociadas a la imagen CRA no se emitirán y deberían eliminarse del DPB. Es probable que dichas situaciones se produzcan en puntos de empalme, donde dos flujos de bits se unen entre sí 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 del DPB con retardo asociadas de picA se pueden eliminar del DPB antes de que puedan emitirse, porque si las imágenes del DPB con retardo asociadas continúan ocupando el DPB, la descodificación de las imágenes comenzando a partir de picA se puede volver problemática, por ejemplo, debido al desbordamiento del búfer. En este caso, el valor de no\_output\_of\_prior\_pics\_flag (por ejemplo, un indicador que indica que las imágenes que se descodificaron y se almacenaron previamente en el DPB se deben eliminar del DPB sin emitirse si se establece en 1, e indica que las imágenes que se descodificaron y se almacenaron previamente en el DPB no deben eliminarse del DPB sin emitirse si se establece en 0) asociado con picA se debe establecer igual a un valor de "1" por el codificador o empalmador, o NoOutputOfPriorPicsFlag (por ejemplo, un valor derivado que se puede determinar en base a la información incluida en el flujo de bits) se puede derivar para que sea igual a un valor de "1" por el descodificador, para descargar las imágenes con retardo sin salida del DPB. Esta operación de empalme se describe con más detalle a continuación con referencia a la **FIG. 4**.

**[0138]** Este proceso de eliminación de imágenes del DPB con retardo asociadas del DPB sin emitirse se puede denominar "descarga". 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 descodificador descargará las imágenes con retardo del DPB asociadas de la imagen IRAP.

### **Flujo de bits incluyendo un punto de empalme**

**[0139]** 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 multicapa 400 creado por 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 de EL 412A, y la BL 410B incluye una imagen de BL 412B. La EL 420A incluye las imágenes de EL 422A, 424A y 426A, y la BL 420B incluye las imágenes de BL 422B, 424B y 426B. El flujo de bits multicapa 400 incluye además las unidades de acceso (AU) 430-460. La AU 430 incluye la imagen de EL 412A y la imagen de BL 412B, la AU 440 incluye la imagen de EL 422A y la imagen de BL 422B, la AU 450 incluye la imagen de EL 424A y la imagen de BL 424B, y la AU 460 incluye la imagen de EL 426A y la imagen de BL 426B. En el ejemplo de la **FIG. 4**, la imagen de BL 422B es una imagen IRAP, y la imagen de 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 IRAP AU no alineada. Además, se debe tener en cuenta que la AU 440 es una unidad de acceso que sigue inmediatamente un punto de empalme 470.

[0140] Aunque el ejemplo de la **FIG. 4** ilustra un caso donde 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 parte del flujo de bits. Por ejemplo, un flujo de bits puede tener las partes A, B y C, estando la parte B entre las partes A y C. Si la parte B se elimina del flujo de bits, las partes restantes A y C se pueden unir entre sí, y el punto en el que se unen entre sí se puede denominar un punto de empalme. Más en general, se puede considerar que un punto de empalme como se analiza en la presente solicitud está presente cuando uno o más parámetros o indicadores señalizados u derivados tienen valores predeterminados. Por ejemplo, sin recibir una indicación específica de que existe un punto de empalme en una localización particular, un descodificador puede determinar el valor de un indicador (por ejemplo, NoCrasOutputFlag) y realizar una o más técnicas descritas en la presente aplicación en base al valor del indicador.

#### **Descarga de imágenes en un contexto multicapa**

[0141] El proceso de descarga de imágenes también es relevante en flujos de bits multicapa. Más específicamente, es relevante 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 describe anteriormente, en algunas implementaciones existentes tales como SHVC y MV-HEVC, una unidad de acceso IRAP se puede definir como una unidad de acceso que contiene una imagen IRAP que tiene nuh\_layer\_id igual a un valor de "0" (independientemente de si otras imágenes en 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).

[0142] Con la posibilidad de tener imágenes IRAP no alineadas en las 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 previa en el contexto de HEVC se pueden producir en diferentes capas de un flujo de bits de SHVC/MV-HEVC. Por ejemplo, una imagen CRA picA 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 una unidad de acceso IRAP inicial que no tiene una imagen CRA en la misma capa que picA. Además, el cambio de resolución de una imagen se podría producir en las 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 del DPB.

#### **Descarga de imágenes en SVC y MVC**

[0143] Debido al diseño de codificación de bucle único de SVC, solo se inserta una imagen reconstruida por unidad de acceso en el DPB, excepto en los casos en los que se usa la denominada escalabilidad granular media (MGS) (en cuyo caso puede haber dos imágenes descodificadas de las denominadas unidades de acceso de imagen clave que se almacenan en el DPB). Sin embargo, en cada unidad de acceso solo se puede emitir la imagen descodificada de la capa más alta. Por lo tanto, las operaciones para gestionar el DPB, incluyendo la descarga de imágenes, solo se refieren a las imágenes en la capa más alta, principalmente porque no se requiere que una imagen descodificada de una capa base esté presente en el DPB para predecir la capa de mejora.

[0144] En MVC, más de una vista puede ser la vista de salida objetivo, y los componentes de la vista descodificados se deben mantener para predecir los componentes de la vista en otra capa, incluso si no son necesarios para predecir los componentes de la vista en la misma capa. Por lo tanto, los componentes de la vista de más de una vista pueden estar presentes en el DPB. El indicador no\_output\_of\_prior\_pics\_flag se señala para cada componente de la vista IDR (por ejemplo, un componente de la vista IDR de una vista no base se señala con non\_idr\_flag igual a un valor de "0"), y la descarga de los componentes de la vista es específica de la capa (o específica 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 la vista en una unidad de acceso es un componente de la vista IDR, todos los componentes de la vista en esa unidad de acceso también son componentes de la vista IDR. Por lo tanto, la operación de descarga también se realiza en todas las vistas en el flujo de bits, aunque la operación puede ser específica de la vista/capa.

**Descarga de imágenes en SHVC y MV-HEVC**

[0145] Cuando se produce una descarga en el diseño actual en SHVC y MV-HEVC, todas las imágenes en el DPB se eliminan sin emitirse (por ejemplo, visualizarse). No es posible que las imágenes una sola capa en el flujo de bits (excepto en el caso trivial cuando solo está presente la capa base en el flujo de bits) se descarguen, por lo tanto, la descarga no es específica de la capa.

**Conformidad de temporización de salida**

[0146] En algunas implementaciones (por ejemplo, SHVC, MV-HEVC, etc.), la salida y eliminación de imágenes del DPB para la conformidad de la temporización de salida se realiza como se describe a continuación. Las partes relevantes para el proceso de descarga 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.

**C.3.2 Eliminación de imágenes del DPB**

[0147] La eliminación de imágenes del DPB antes de la descodificación de la imagen actual (pero después de analizar sintácticamente la cabecera de fragmento del primer fragmento de la imagen actual) sucede instantáneamente en el momento de la eliminación del CPB de la primera unidad de descodificación de la unidad de acceso n (que contiene la imagen actual) y procede como sigue:

- Se invoca el proceso de descodificación para RPS como se especifica en la subcláusula 8.3.2.
- *Cuando la imagen actual es una imagen IRAP con NoRaslOutputFlag igual a 1 que no es la imagen 0, se aplican las siguientes etapas ordenadas:*

1. *La variable NoOutputOfPriorPicsFlag se deriva para el descodificador bajo prueba como sigue:*

- *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 se puede (pero no debe) establecer a 1 por el descodificador 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 descodificador bajo prueba puede establecer NoOutputOfPriorPicsFlag 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 descodificador bajo prueba se aplica para el HRD, de modo que cuando el valor de NoOutputOfPriorPicsFlag es igual a 1, todos los búferes de almacenamiento de imágenes en el DPB se vacían sin emitir las imágenes que contienen, y la plenitud del DPB se establece igual a 0.*

- Cuando se cumplen las dos condiciones siguientes para cualquier imagen k en el DPB, todas esas imágenes k en el DPB se eliminan del DPB:
  - la imagen k está marcada como "no usada como referencia"
  - la imagen k tiene PicOutputFlag igual a 0 o su tiempo de salida del DPB es menor o igual que el tiempo de eliminación del CPB de la primera unidad de descodificación (indicada como unidad de descodificación m) de la imagen actual n; es decir, DpbOutputTime[k] es menor o igual que CpbRemovalTime(m)
- Para cada imagen que se elimina del DPB, la plenitud del DPB se decrementa en uno.

**F.13.3.2 Eliminación de imágenes del DPB**

[0148] Las especificaciones de la subcláusula C.3.2 se aplican por separado para cada conjunto de imágenes descodificadas con un valor particular de nuh\_layer\_id con las siguientes modificaciones. - Sustituir "La eliminación

de imágenes del DPB antes de descodificar la imagen actual (pero después de analizar sintácticamente la cabecera de fragmento del primer fragmento de la imagen actual) sucede instantáneamente en el momento de eliminación del CPB de la primera unidad de descodificación de la unidad de acceso n (que contiene la imagen actual) y procede de la siguiente manera:" por "La eliminación de imágenes del DPB antes de descodificar la imagen actual (pero después de analizar sintácticamente la cabecera de fragmento del primer fragmento de la imagen actual) sucede instantáneamente en el momento de eliminación del CPB de la primera unidad de descodificación de la imagen n y procede de la siguiente manera:".

### **Conformidad de orden de salida**

[0149] En algunas implementaciones (por ejemplo, SHVC, MV-HEVC, etc.), la salida y eliminación de imágenes del DPB para la conformidad de orden de salida se realiza como se describe a continuación. Las partes relevantes para el proceso de descarga se muestran en *cursiva*. En el ejemplo a continuación, la eliminación de imágenes, cuando se invoca, se realiza para todas las capas.

#### **F.13.5.2.2 Salida y eliminación de imágenes del DPB**

[0150] La salida y eliminación de imágenes del DPB antes de descodificar la imagen actual (pero después de analizar sintácticamente la cabecera de fragmento del primer fragmento de la imagen actual) sucede instantáneamente cuando la primera unidad de descodificación de la imagen actual se elimina del CPB y procede de la siguiente manera:

Se invoca el proceso de descodificación para RPS como se especifica en la subcláusula F.8.3.2.

- Si la imagen actual es una imagen IRAP con NoRaslOutputFlag 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 descodificador bajo prueba como sigue:

- 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 se puede (pero no debe) establecer en 1 por el descodificador 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 descodificador bajo prueba puede establecer NoOutputOfPriorPicsFlag 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 descodificador bajo prueba se aplica para el HRD como sigue:

- Si NoOutputOfPriorPicsFlag es igual a 1, todos los búferes de almacenamiento de imágenes en el DPB se vacían sin emitir las imágenes que contienen, y la plenitud del DPB se establece en 0.
- De otro modo (NoOutputOfPriorPicsFlag es igual a 0), todos los búferes de almacenamiento de imágenes que contienen una imagen que está marcada como "no necesaria para la salida" y "no usada como referencia" se vacían (sin salida), y todos los búferes de almacenamiento de imágenes no vacíos en el DPB se vacían invocando repetidamente el proceso de "expulsión" especificado en la subcláusula F.13.5.2.4, y la plenitud del DPB se establece igual a 0.

- De otro modo (la imagen actual no es una imagen IRAP con NoRaslOutputFlag igual a 1 o con nuh\_layer\_id distinto de 0), todos los búferes de almacenamiento de imágenes que contienen una imagen que está marcada como "no necesaria para la salida" y "no usada como referencia" se vacían (sin salida). Para cada búfer de almacenamiento de imágenes que se vacía, la plenitud del DPB se decrementa en uno. La variable currLayerId se establece igual a nuh\_layer\_id de la imagen descodificada actual y cuando una o más de las siguientes condiciones son verdaderas, el proceso de "expulsión" especificado en la subcláusula F.13.5.2.4 se invoca repetidamente mientras se decrementa aún más la plenitud del DPB en uno para cada búfer de almacenamiento de imágenes adicional que se vacía, hasta que ninguna de las siguientes condiciones es verdadera:

- El número de imágenes con `nuh_layer_id` igual a `currLayerId` en el DPB que están marcadas como "necesarias para la salida" es mayor que `sps_max_num_reorder_pics[HighestTid]` del SPS activo (cuando `currLayerId` es igual a 0) o del 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 del SPS de la capa activa si el valor de `currLayerId` no es igual a 0 y hay al menos una imagen con `nuh_layer_id` igual a `currLayerId` en el DPB que está marcada como "necesaria para la salida" para la cual la variable asociada `PicLatencyCount[currLayerId]` es mayor o igual que `SpsMaxLatencyPictures[HighestTid]` derivada del SPS activo (cuando `currLayerId` es igual a 0) o del SPS de la capa activa para el valor de `currLayerId`.
- El número de imágenes con `nuh_layer_id` igual a `currLayerId` en el DPB es mayor o igual que `sps_max_dec_pic_buffering_minus1[HighestTid] + 1` del SPS activo (cuando `currLayerId` es igual a 0) o del SPS de la capa activa para el valor de `currLayerId`.

#### **Comparación de la conformidad de temporización de salida y la conformidad de orden de salida**

[0151] Como se describe anteriormente, la conformidad de temporización de salida y la conformidad de orden de salida pueden no dar como resultado el mismo comportamiento de descarga. Por ejemplo, para los descodificadores con conformidad de temporización de salida, se invoca la descarga 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 la descarga, se descargan todas las imágenes descodificadas de esa capa en el DPB. Por otro lado, para descodificadores con conformidad de orden de salida, la descarga 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 descarga, se descargan todas las imágenes descodificadas de todas las capas en el DPB.

[0152] 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 IRAP AU activa una resolución diferente, y la resolución de la BL no puede cambiar en esta AU debido a que la imagen de BL está en una AU no IRAP (por ejemplo, no es una imagen IRAP), se puede desear una descarga de imágenes específica de la capa. Aquí, solo se descargarán las imágenes de la EL, pero no de la BL. Esta característica no está disponible para la conformidad de orden de salida.

[0153] En un flujo de bits que tiene dos capas, en un caso en el que una unidad de acceso incluye una imagen de BL que es una imagen IDR y una imagen de EL que es una imagen no IRAP, la resolución de la imagen de BL se puede actualizar en la unidad de acceso, mientras que la resolución de la imagen de EL no se actualiza. En dicho caso, la descarga se debe realizar solo para las imágenes de la BL, y las imágenes de la EL no se deben descargar. Esta característica no está disponible para la conformidad de orden de salida.

#### **Activación de la conformidad de temporización de salida y la conformidad de orden de salida del SPS**

[0154] Cuando un SPS con `nuh_layer_id` igual a un valor de "0" está presente en una unidad de acceso después de la última unidad VCL NAL que tiene `nuh_layer_id` igual a un valor de "0" en un flujo de bits que tiene más de una capa y la siguiente unidad de acceso en orden de descodificación tiene un delimitador de unidad de acceso (por ejemplo, que indica que hay unidades VCL NAL adicionales que tienen valores de `nuh_layer_id` mayores que 0 presentes en el flujo de bits), entonces el flujo de bits extraído con el conjunto de capas de salida que solo contiene la BL puede no ser conforme. Por ejemplo, el flujo de bits extraído puede incluir una unidad SPS NAL después de todas sus unidades VCL NAL. Dicha unidad SPS NAL se puede denominar SPS sin referencia. Los descodificadores típicamente esperan procesar unidades VCL NAL adicionales que siguen en el flujo de bits después de procesar una unidad SPS NAL. Por tanto, en algunos esquemas de codificación, dicho SPS sin referencia puede dar como resultado un flujo de bits no conforme.

#### **Modos de realización de ejemplo**

[0155] A continuación se describirán varios procedimientos que se pueden usar para abordar ciertos problemas descritos anteriormente. Algunos de estos procedimientos se pueden aplicar de forma independiente y algunos de ellos se pueden aplicar en combinación. Además, la sintaxis y la semántica de ejemplo que se pueden usar para implementar uno o más de los procedimientos descritos en el presente documento también se proporcionan a continuación. Cuando ciertas partes de la especificación de HEVC se reproducen para ilustrar las adiciones y eliminaciones que se pueden incorporar para implementar uno o más de los procedimientos descritos en el presente documento, dichas modificaciones se muestran en *cursiva*.

#### **Descarga de imágenes específica de la capa**

[0156] En algunos modos de realización, la descarga de imágenes se realiza de una manera específica de la capa para ambos tipos de conformidad del descodificador (por ejemplo, conformidad de temporización de salida y conformidad de orden de salida). El proceso de descarga se puede producir (o se puede habilitar para que se

produzca) en cada imagen IRAP con NoRasOutputFlag igual a un valor de "1" y en cada imagen LIP (por ejemplo, en lugar de producirse solo en una imagen IRAP que tiene NoRasOutputFlag igual a un valor de "1" y nuh\_layer\_id igual a un valor de "0").

## 5 **Señalización del indicador que indica la salida de imágenes anteriores**

[0157] En algunos modos de realización, el indicador no\_output\_of\_prior\_pics\_flag se señala para todas las imágenes IRAP en la BL (por ejemplo, teniendo nuh\_layer\_id igual a un valor de "0"), y el indicador no\_output\_of\_prior\_pics\_flag se señala en la cabecera de segmento de fragmento de todas las unidades VCL NAL que tienen un nuh\_layer\_id que no es igual a un valor de "0". Como se analiza anteriormente, no\_output\_of\_prior\_pics\_flag puede indicar si las imágenes que se descodificaron y almacenaron previamente en el DPB se deben eliminar del DPB sin emitirse. En otros modos de realización, el indicador no\_output\_of\_prior\_pics\_flag se señala en la cabecera de segmento de fragmento de todas las unidades VCL NAL.

[0158] Si la imagen actual está en la EL (por ejemplo, tiene un nuh\_layer\_id mayor que un valor de "0"), las condiciones que normalmente se pueden comprobar (por ejemplo, si la imagen actual es una imagen IRAP) antes de que se señalice no\_output\_of\_prior\_pics\_flag se pueden omitir. Por ejemplo, el indicador no\_output\_of\_prior\_pics\_flag se puede señalar para cada EL presente en el flujo de bits. El indicador no\_output\_of\_prior\_pics\_flag puede estar presente en la posición original en la tabla de sintaxis (por ejemplo, sin la etapa adicional de comprobar si se cumplen las condiciones). Si la imagen actual es una imagen de BL (por ejemplo, tiene un nuh\_layer\_id igual a un valor de "0") que no es una imagen IRAP, el indicador no\_output\_of\_prior\_pics\_flag (u otro indicador que tenga una indicación y/o función similares) puede estar presente como uno de los bits reservados en la cabecera de fragmento o como parte de la extensión de la cabecera de fragmento. Si la imagen actual es una imagen de BL (por ejemplo, tiene un nuh\_layer\_id igual a un valor de "0") y es una imagen IRAP, la señalización del indicador no\_output\_of\_prior\_pics\_flag puede permanecer sin cambios.

## **Salida de imágenes en base a las condiciones de la unidad de acceso**

[0159] En algunos modos de realización, la variable NoOutputOfPriorPicsFlag (por ejemplo, un valor derivado por el descodificador para determinar si se emiten o no las imágenes en el DPB antes de que se descargue el DPB) se deriva en base a no\_output\_of\_prior\_pics\_flag y otras condiciones, al menos para todas las imágenes LIP que no son imágenes IRAP. 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 en base a la información incluida en el flujo de bits. Un descodificador puede derivar el valor de NoOutputOfPriorPicsFlag en base al valor de no\_output\_of\_prior\_pics\_flag y otras condiciones, y a continuación usar el valor derivado de NoOutputOfPriorPicsFlag para determinar si se emiten las imágenes o no. En algunos modos de realización, el valor de NoOutputOfPriorPicsFlag para cada imagen LIP picA que no es una imagen IRAP se puede inferir en base al valor de NoCIRasOutputFlag asociado a la imagen IRAP que pertenece a la unidad de acceso que contiene picA, que tiene nuh\_layer\_id igual a un valor de "0", y que tiene NoRasOutputFlag igual a un valor de "1". En algunos modos de realización, 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.

[0160] En algunos modos de realización, NoCIRasOutputFlag y NoRasOutputFlag pueden ser variables derivadas en base a la información incluida en el flujo de bits. Por ejemplo, NoRasOutputFlag se puede derivar para cada imagen IRAP (por ejemplo, en la BL y/o la EL), y NoCIRasOutputFlag se puede derivar solo para las imágenes de capa más baja (por ejemplo, imágenes de BL). El valor de cada uno de NoCIRasOutputFlag y NoRasOutputFlag puede indicar que algunas imágenes en el flujo de bits pueden no ser descodificables correctamente debido a la falta de disponibilidad de ciertas imágenes de referencia. Dicha falta de disponibilidad de imágenes de referencia se puede producir en puntos de acceso aleatorio. Las imágenes de omisión de acceso aleatorio de capas cruzadas (CL-RAS) son, de alguna manera, el equivalente multicapa de las imágenes RASL. Si un descodificador comienza a descodificar 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 de EL en la unidad de acceso no es una imagen IRAP, entonces esa imagen de EL es una imagen CL-RAS. Todas las imágenes en la EL pueden ser imágenes CL-RAS (por ejemplo, descodificables, pero no correctamente descodificables) 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 la EL se ha inicializado.

[0161] Por ejemplo, en el ejemplo de la FIG. 4, la imagen de EL 422A puede ser una imagen LIP que no es una imagen IRAP, y la imagen de BL 422B puede ser una imagen IRAP que tiene un indicador NoCIRasOutputFlag asociado a la misma. En este ejemplo, el valor de NoOutputOfPriorPicsFlag asociado a la imagen de EL 422A se puede inferir en base al valor de NoCIRasOutputFlag asociado a la imagen de BL 422B. Por ejemplo, si NoCIRasOutputFlag es igual a un valor de "1", NoOutputOfPriorPicsFlag para la imagen de EL 422A también se puede establecer en un valor de "1", lo que hace que las imágenes en el DPB no se emitan antes de que se eliminen del DPB. Por otro lado, si NoCIRasOutputFlag es igual a un valor de "0", NoOutputOfPriorPicsFlag para la imagen de EL 422A también se puede establecer en un valor de "0", lo que hace que las imágenes en el DPB

se eliminan del DPB después de emitirse.

#### **Diagrama de flujo de ejemplo para la emisión de imágenes en base a las condiciones de la unidad de acceso**

[0162] Con referencia a la FIG. 5, se describirá una rutina de ejemplo para descargar el DPB. La FIG. 5 es un diagrama de flujo que ilustra un procedimiento 500 para codificar información de vídeo, de acuerdo con un modo de realización de la presente divulgación. Las etapas ilustradas en la FIG. 5 se pueden realizar por un codificador (por ejemplo, el codificador de vídeo como se muestra en la FIG. 2A o la FIG. 2B), un decodificador (por ejemplo, el decodificador de vídeo como se muestra en la FIG. 3A o la FIG. 3B) o cualquier otro componente. Por conveniencia, el procedimiento 500 se describe como realizado por un codificador, que puede ser el codificador, el decodificador u otro componente.

[0163] El procedimiento 500 comienza en el bloque 501. En el bloque 505, el codificador determina si una imagen es una imagen no IRAP de punto de empalme. Por ejemplo, el codificador puede determinar si la imagen es una imagen no IRAP que está en una unidad de acceso que sigue inmediatamente a un punto de empalme. En algunos modos de realización, si una imagen particular está en una unidad de acceso que sigue inmediatamente a un punto de empalme, se puede señalar o procesar como un indicador. En dichos modos de realización, un valor del indicador de 1 puede indicar que la imagen está en una unidad de acceso que sigue inmediatamente a un punto de empalme, y un valor del indicador de 0 puede indicar que la imagen no está en una unidad de acceso que sigue inmediatamente a un punto de empalme. Si el codificador determina que la imagen no es una imagen no IRAP de punto de empalme, el procedimiento 500 continúa en el bloque 510. Si el codificador determina que la imagen es una imagen no IRAP de punto de empalme, el procedimiento 500 continúa en el bloque 515.

[0164] En el bloque 510, el codificador emite las imágenes en el DPB antes de eliminar las imágenes del DPB. En el bloque 515, el codificador elimina las imágenes en el DPB sin emitir las imágenes. El procedimiento 500 termina en 515.

[0165] Como se analiza anteriormente, uno o más componentes del codificador de vídeo 20 de la FIG. 2A, el codificador de vídeo 23 de la FIG. 2B, el decodificador de vídeo 30 de la FIG. 3A, o el decodificador de vídeo 33 de la FIG. 3B (por ejemplo, la unidad de predicción entre capas 128 y/o la unidad de predicción entre capas 166) se pueden usar para implementar cualquiera de las técnicas analizadas en la presente divulgación, tales como determinar si la imagen es una imagen no IRAP de punto de empalme y emitir las imágenes y/o eliminar las imágenes del DPB.

#### **Combinación de procedimientos de descarga del DPB**

[0166] En algunos modos de realización, el proceso de descarga es específico de la capa solo cuando se invoca en una de las imágenes de EL que también son imágenes IRAP. Cuando el proceso de descarga se invoca en una imagen IRAP que pertenece a la BL que tiene NoRasOutputFlag igual a un valor de "1", todas las imágenes de todas las capas se pueden descargar del DPB.

#### **Unidades y conjuntos de parámetros de capa de abstracción de red (NAL)**

[0167] Como se analiza anteriormente, los parámetros usados por un codificador o un decodificador se pueden agrupar en conjuntos de parámetros en base al nivel de codificación en el que se pueden usar. Por ejemplo, los parámetros que se usan por una o más secuencias de vídeo codificado en el flujo de bits se pueden incluir en un conjunto de parámetros de vídeo (VPS), y los parámetros que se usan por una o más imágenes en una secuencia de vídeo codificado se pueden incluir en un conjunto de parámetros de secuencia (SPS). De forma similar, los parámetros que se usan por uno o más fragmentos en una imagen se pueden incluir en un conjunto de parámetros de imagen (PPS), y otros parámetros que son específicos de un único fragmento se pueden incluir en una cabecera de fragmento. Dichos conjuntos de parámetros se pueden activar (o indicar como activos) para una capa dada por unidades NAL de conjuntos de parámetros (por ejemplo, unidad SPS NAL, unidad PPS NAL, etc.). Una unidad NAL comprende una carga útil de secuencia de bytes sin procesar (RBSP) y una cabecera de unidad NAL. La RBSP puede especificar una ID de conjunto de parámetros (por ejemplo, SPS ID), y la cabecera de unidad NAL puede especificar la ID de capa, que puede indicar qué capas pueden usar el SPS.

[0168] En algunos casos, puede ser beneficioso proporcionar múltiples instancias en el flujo de bits de un conjunto de parámetros particular que se puede usar o activar para una capa dada. Por ejemplo, incluso después de que el conjunto de parámetros particular ya se haya activado para la capa dada, se puede proporcionar una instancia adicional del conjunto de parámetros particular en el flujo de bits para su uso por la capa dada. Cuando el flujo de bits contiene dicha instancia adicional de un SPS, incluso si se pierde el conjunto de parámetros particular señalado previamente, un decodificador puede usar el conjunto de parámetros particular señalado posteriormente para la capa dada.

[0169] Sin embargo, en algunos esquemas de codificación, cuando se proporciona dicha instancia adicional de un conjunto de parámetros (por ejemplo, SPS repetido) en el flujo de bits, se puede requerir que el contenido del

conjunto de parámetros (por ejemplo, una unidad SPS NAL) sea idéntico a todas las instancias previas del conjunto de parámetros. Por ejemplo, si un flujo de bits comprende una capa base y una capa de mejora, después de que se proporcione una unidad SPS NAL en el flujo de bits, tanto la capa base como la capa de mejora se pueden referir al SPS. Puede haber una situación en la que, después de que se hayan proporcionado todas las unidades NAL (por ejemplo, unidades VCL NAL) de la capa base en el flujo de bits, sea deseable proporcionar el SPS de nuevo en el flujo de bits, por ejemplo, para mejorar la recuperación frente a errores. En dicha situación, se puede requerir que la unidad SPS NAL posterior tenga el mismo contenido que la unidad SPS NAL que se proporcionó previamente en el flujo de bits. Debido a que las unidades SPS NAL pueden especificar una ID de capa para indicar qué capas pueden usar las unidades SPS NAL, en el ejemplo descrito anteriormente, se puede requerir por la restricción del flujo de bits que la unidad SPS NAL posterior especifique una ID de capa que sea la misma que la unidad SPS NAL proporcionada previamente, lo que puede indicar que tanto la capa base como la capa de mejora pueden usar el SPS, aunque la unidad SPS NAL posterior se puede usar únicamente por la capa de mejora. Si ambas unidades SPS NAL especificaron la misma ID de capa mientras se usaban por capas diferentes, pueden surgir problemas durante el proceso de descodificación.

**[0170]** Por ejemplo, si un flujo de bits particular tiene una capa base y una capa de mejora, donde tanto la capa base como la capa de mejora se refieren al mismo SPS. Si la EL tiene una velocidad de tramas mayor que la BL, las últimas unidades de acceso en el flujo de bits pueden contener solo imágenes de EL y no imágenes de BL. En dicho ejemplo, si una de las últimas unidades de acceso contenía una unidad SPS NAL repetida, la restricción del flujo de bits descrita anteriormente puede obligar a que la ID de capa de la unidad SPS NAL sea la misma que una unidad SPS NAL previa que activa el SPS (por ejemplo, un codificador puede determinar que dicha restricción del flujo de bits es aplicable y atenerse a la restricción del flujo de bits de modo que el flujo de bits codificado sea conforme a la restricción del flujo de bits). Por ejemplo, dicha unidad SPS NAL previa se puede usar por la capa base, y la unidad SPS NAL previa puede tener un valor de la ID de capa de 0, lo que indica que la capa base puede usar el SPS. En dicho caso, la ID de capa de la unidad SPS NAL repetida también tendría que ser igual a un valor de "0", aunque la unidad SPS NAL repetida no esté destinada a usarse por la capa base. Si un descodificador intenta extraer la capa base del flujo de bits en este ejemplo (por ejemplo, tomando todas las unidades NAL que tienen una ID de capa de 0), el flujo de bits resultante tendría la unidad SPS NAL repetida al final del flujo de bits. Esto puede ser problemático porque el descodificador puede suponer, tras procesar la unidad SPS NAL repetida, que la unidad SPS NAL repetida señala el comienzo de la siguiente unidad de acceso (o secuencia de vídeo codificada). Para evitar dicho problema, el codificador puede decidir no proporcionar la unidad SPS NAL posterior en el flujo de bits, renunciando de este modo a los beneficios potenciales asociados a las unidades SPS NAL repetidas.

#### **Activación de la carga útil de la secuencia de bytes sin procesar (RBSP) del conjunto de parámetros de secuencia (SPS)**

**[0171]** Una SPS RBSP incluye parámetros a los que puede hacer referencia una o más (PPS) RBSP de una o más unidades SEI NAL que contienen un mensaje SEI de conjuntos de parámetros activos. Inicialmente, se puede considerar que cada SPS RBSP no está activa para ninguna capa al comienzo del proceso de descodificación. Para cada capa, como máximo se considera activa una SPS RBSP en cualquier momento dado durante el proceso de descodificación, y la activación de cualquier SPS RBSP particular para una capa particular da como resultado la desactivación de la SPS RBSP activa previamente para esa capa particular, si existía alguna.

**[0172]** La misma SPS RBSP puede ser la SPS RBSP activa para más de una capa. Por ejemplo, si una capa base y una capa de mejora contienen una imagen que se refiere a un PPS que a su vez se refiere a un SPS que tiene una SPS ID de 3, el SPS que tiene una SPS ID de 3 es la SPS RBSP activa para la capa de referencia y la capa de mejora.

**[0173]** Cuando una SPS RBSP (por ejemplo, que tiene una SPS ID particular) ya no está activa para una capa no base particular (por ejemplo, que tiene un valor de ID de capa distinto de cero o una ID de capa mayor que 0) que tiene una ID de capa (por ejemplo, nuh\_layer\_id) de X, y se hace referencia a la SPS RBSP en la RBSP de un conjunto de parámetros de imagen (PPS), la SPS RBSP se activa para la capa no base particular. Este SPS se puede denominar la SPS RBSP activa para la capa no base particular hasta que se desactiva por la activación de otra SPS RBSP para la capa no base particular.

#### **Activación de conjuntos de parámetros**

**[0174]** Como se analiza anteriormente, en algunos esquemas de codificación, una imagen de inicialización de capa ("imagen LIP") se define como (1) una imagen que es una imagen IRAP que tiene un indicador NoRasOutputFlag (por ejemplo, un indicador que indica si se deben emitir las imágenes RASL) establecido en 1 o (2) una imagen que está contenida en 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 NoRasOutputFlag establecido en 1.

**[0175]** En algunos modos de realización, se puede activar un SPS en cada imagen LIP. Por ejemplo, cada

imagen IRAP que tiene un indicador NoRaslOutputFlag establecido en 1 o cada imagen que está contenida en una unidad de acceso IRAP inicial, puede activar un nuevo SPS, que puede ser el mismo o diferente del SPS que estaba activado previamente (por ejemplo, teniendo diferentes parámetros tales como el tamaño de la imagen, etc.). Además, después de que se haya activado el SPS, se puede proporcionar una instancia adicional del mismo SPS en el flujo de bits. Dicha instancia repetida de un SPS (o SPS repetido) puede mejorar la recuperación frente a errores sirviendo como un SPS de reserva en el caso de que el SPS proporcionado previamente se pierda o se elimine del flujo de bits.

#### **Diagrama de flujo de ejemplo para proporcionar un SPS repetido en el flujo de bits**

[0176] La FIG. 6 es un diagrama de flujo que ilustra un procedimiento 600 para codificar información de vídeo, de acuerdo con un modo de realización de la presente divulgación. Las etapas ilustradas en la FIG. 6 se pueden realizar por un codificador (por ejemplo, el codificador de vídeo como se muestra en la FIG. 2A o la FIG. 2B), un descodificador (por ejemplo, el descodificador de vídeo como se muestra en la FIG. 3A o la FIG. 3B) o cualquier otro componente. Por conveniencia, el procedimiento 600 se describe como realizado por un codificador, que puede ser el codificador, el descodificador u otro componente.

[0177] El procedimiento 600 comienza en el bloque 601. En el bloque 605, el codificador proporciona un conjunto de parámetros de secuencia (SPS) en un flujo de bits, con una indicación de que el SPS se puede activar para una primera capa de vídeo y una segunda capa de vídeo. Por ejemplo, la primera capa de vídeo puede ser una capa base y la segunda capa de vídeo puede ser una capa de mejora. La segunda capa de vídeo puede ser cualquier capa que tenga una ID de capa diferente a la de la primera capa de vídeo. El SPS se puede proporcionar en el flujo de bits en forma de una unidad SPS NAL que tiene una ID de capa y una ID de SPS. Por ejemplo, la unidad SPS NAL puede tener una ID de capa que indica que el SPS se puede activar para la primera y la segunda capas de vídeo. En algunos modos de realización, si la ID de capa de un SPS tiene un valor de 0, el SPS se puede activar para cualquier capa que tenga una ID de capa mayor o igual que un valor de "0". Por ejemplo, en un caso donde la capa base tiene una ID de capa de 0 y la capa de mejora tiene una ID de capa de 1, si la ID de capa de un SPS tiene un valor de 0, el SPS se puede activar por la capa base y la capa de mejora.

[0178] En el bloque 610, el codificador proporciona el mismo SPS (por ejemplo, un SPS repetido, que es una unidad SPS NAL que tiene la misma SPS ID que la unidad SPS NAL proporcionada previamente) en el flujo de bits con una indicación de que el SPS se puede activar para la segunda capa de vídeo, pero no para la primera capa de vídeo. Por ejemplo, la unidad SPS NAL repetida puede tener una ID de capa que sea diferente a la de la unidad SPS NAL proporcionada previamente. En el caso de que la capa base tenga una ID de capa de 0 y la capa de mejora tenga una ID de capa de 1, si la ID de capa del SPS repetido tiene un valor de 1, el SPS repetido se puede activar por la capa de mejora (por ejemplo, que tiene un valor de la ID de capa de 1) pero no por la capa base (por ejemplo, que tiene un valor de la ID de capa de 0). Proporcionar un SPS repetido que tenga la misma SPS ID que un SPS proporcionado previamente, pero una ID de capa diferente puede ser útil si el SPS repetido se proporciona en el flujo de bits después de que se hayan proporcionado todas las unidades VCL NAL de una o más capas inferiores (por ejemplo, la capa base). Al proporcionar un SPS repetido que tiene una ID de capa que indica que el SPS repetido no se va a activar para la una o más capas inferiores, cuando la una o más capas inferiores se extraen del flujo de bits, el flujo de bits extraído resultante no incluiría el SPS repetido, que no es necesario para las capas inferiores porque todas las unidades VCL NAL se han proporcionado antes del SPS repetido. El procedimiento 600 termina en 615.

[0179] Como se analiza anteriormente, uno o más componentes del codificador de vídeo 20 de la FIG. 2A, el codificador de vídeo 23 de la FIG. 2B, el descodificador de vídeo 30 de la FIG. 3A, o el descodificador 33 de vídeo de la FIG. 3B (por ejemplo, la unidad de predicción entre capas 128 y/o la unidad de predicción entre capas 166) se pueden usar para implementar cualquiera de las técnicas analizadas en la presente divulgación, tal como proporcionar un conjunto de parámetros de secuencia (SPS) en un flujo de bits, con una indicación de que el SPS se puede activar para una primera capa de vídeo y una segunda capa de vídeo, y proporcionar el mismo SPS en el flujo de bits con una indicación de que el SPS se puede activar para la segunda capa de vídeo pero no para la primera capa de vídeo.

[0180] En el procedimiento 600, uno o más de los bloques mostrados en la FIG. 6 se pueden eliminar (por ejemplo, no realizar) y/o el orden en el que se realiza el procedimiento se puede cambiar. En algunos modos de realización, se pueden añadir bloques adicionales al procedimiento 600. Aunque el procedimiento 600 se describe con referencias a un SPS, se debe apreciar que las técnicas descritas en relación con el procedimiento 600 se pueden extender y aplicar a otros conjuntos de parámetros tales como el VPS, el PPS y la cabecera de fragmento.

#### **Restricción del flujo de bits con respecto a SPS repetidos**

[0181] En algunos modos de realización, una restricción del flujo de bits puede especificar que cuando una unidad SPS NAL (por ejemplo, SPS repetido) contiene el mismo valor de SPS ID (por ejemplo, sps\_seq\_parameter\_set\_id) que un SPS señalizado previamente, entonces la SPS RBSP de la unidad SPS NAL repetida tendrá el mismo contenido que el de la unidad SPS NAL señalizada previamente, a menos que el SPS

repetido siga a la última imagen codificada para la que se requiere que el SPS activo permanezca activo y preceda a la primera unidad NAL que activa un SPS con el mismo valor de SPS ID (por ejemplo, `sps_seq_parameter_set_id`).

## 5 Implementación de ejemplo 1

[0182] En los modos de realización descritos a continuación, el valor de `no_output_of_prior_pics_flag` se señala en las cabeceras de segmento de fragmento de todas las unidades VCL NAL. Cuando determinadas partes de la especificación de HEVC se reproducen para ilustrar las adiciones y eliminaciones que se pueden incorporar para implementar uno o más de los procedimientos descritos en el presente documento, dichas modificaciones se muestran en *cursiva*.

### Sintaxis de la cabecera de segmento de fragmento

15 [0183] La siguiente sintaxis de ejemplo se puede usar para implementar uno o más de los modos de realización descritos en el presente documento. Las adiciones al lenguaje existente en la especificación HEVC se muestran en *cursiva*.

<code>slice_segment_header()</code> {	Descriptor
<b><code>first_slice_segment_in_pic_flag</code></b>	<code>u(1)</code>
<i><code>si(nuh_layer_id &gt; 0    ( nal_unit_type &gt;= BLA_W_LP &amp;&amp; nal_unit_type &lt;= RSV_IRAP_VCL23 ))</code></i>	
<b><code>no_output_of_prior_pics_flag</code></b>	<code>u(1)</code>
<b><code>slice_pic_parameter_set_id</code></b>	<code>ue(v)</code>
...	

20 **Tabla 1. Sintaxis de la cabecera de segmento de fragmento de ejemplo**

### Semántica de la cabecera de segmento de fragmento

25 [0184] La siguiente semántica de ejemplo se puede usar para implementar uno o más de los modos de realización descritos en el presente documento. Las adiciones y eliminaciones del lenguaje existente en la especificación de HEVC se muestran en *cursiva*. **`no_output_of_prior_pics_flag`** afecta a la salida de imágenes descodificadas previamente en el búfer de imágenes descodificadas después de la descodificación de una IDR, una imagen BLA o una LIP que no está *contenida en la primera unidad de acceso* en el flujo de bits como se especifica en el Anexo C.

30 [0185] En un modo de realización, `base_no_output_of_prior_pics_flag` se puede señalar para permitir que la imagen de BL que no es una IRAP sea una imagen LIP. ***`base_no_output_of_prior_pics_flag`*** afecta a la salida de imágenes descodificadas previamente en el búfer de imágenes descodificadas después de una imagen de inicialización de capa que no está *contenida en la primera unidad de acceso en el flujo de bits como se especifica en el Anexo C*. Cuando está presente, `no_output_of_prior_pics_flag` se establece para ser igual a `base_no_output_of_pics_flag`.

[0186] En otro modo de realización, no se señala `base_no_output_of_prior_pics_flag`.

## 40 Cambios en el proceso de activación

[0187] El proceso de activación en la especificación de HEVC actual (por ejemplo, la sección F.7.4.2.4.2) se modifica como se muestra a continuación, siendo el resto del proceso el mismo. Las adiciones al lenguaje existente en la especificación HEVC se muestran en *cursiva*.

45 Cualquier unidad SPS NAL que contiene el valor de `sps_seq_parameter_set_id` para la SPS RBSP de capa activa tendrá el mismo contenido de *SPS RBSP* que la SPS RBSP de capa activa a menos que siga a la última imagen codificada para la cual se requiere que el SPS de capa activa esté activo y preceda a la primera unidad NAL que active un SPS del mismo valor de `seq_parameter_set_id`.

50 [0188] En algunos modos de realización, se pueden añadir restricciones similares a otros conjuntos de parámetros, tales como conjuntos de parámetros de vídeo (VPS) y conjuntos de parámetros de imagen (PPS).

### Cambios en la eliminación de imágenes del DPB

55 [0189] El siguiente texto de ejemplo se puede usar para implementar uno o más de los modos de realización descritos en el presente documento. Las adiciones y eliminaciones del lenguaje existente en la especificación de HEVC se muestran en *cursiva*.

### C.3.2 Eliminación de imágenes del DPB

**[0190]** La eliminación de imágenes del DPB antes de la decodificación de la imagen actual (pero después de analizar sintácticamente la cabecera de fragmento del primer fragmento de la imagen actual) sucede instantáneamente en el momento de la eliminación del CPB de la primera unidad de decodificación de la unidad de acceso  $n$  (que contiene la imagen actual) y procede como sigue:

- Se invoca el proceso de decodificación para RPS como se especifica en la subcláusula 8.3.2.
- Si la imagen actual es una *LIP* (o una imagen *IRAP* con *NoRaslOutputFlag* 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 como sigue:

- 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 *de la imagen actual* 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 *con la misma ID de capa que la imagen actual*, *NoOutputOfPriorPicsFlag* se puede (pero no debe) establecer en 1 por 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 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 al HRD, de modo que cuando el valor de *NoOutputOfPriorPicsFlag* es igual a 1, todos los búferes de almacenamiento de imágenes *que están en el DPB y que contienen imágenes que tienen el mismo valor de nuh\_layer\_id que el de la imagen actual*, se vacían sin emitir las imágenes que contienen, y la plenitud del DPB se decrementa en el número de imágenes eliminadas.

- Cuando se cumplen las dos condiciones siguientes para cualquier imagen  $k$  en el DPB, todas esas imágenes  $k$  en el DPB se eliminan del DPB:
  - la imagen  $k$  está marcada como "no usada como referencia"
  - la imagen  $k$  tiene *PicOutputFlag* igual a 0 o su tiempo de salida del DPB es menor o igual que el tiempo de eliminación del 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 del DPB, la plenitud del DPB se decrementa en uno.

### **Cambios en la salida y eliminación de imágenes del DPB**

**[0191]** El siguiente texto de ejemplo se puede usar para implementar uno o más de los modos de realización descritos en el presente documento. Las adiciones y eliminaciones del lenguaje existente en la especificación de HEVC se muestran en *cursiva*.

#### **F.13.5.2.2 Salida y eliminación de imágenes del DPB**

**[0192]** La salida y eliminación de imágenes del DPB antes de decodificar la imagen actual (pero después de analizar sintácticamente la cabecera de fragmento del primer fragmento de la imagen actual) sucede instantáneamente cuando la primera unidad de decodificación de la imagen actual se elimina del CPB y procede de la siguiente manera:

Se invoca el proceso de decodificación para RPS como se especifica en la subcláusula F.8.3.2.

- Si la imagen actual es una *LIP* (o una imagen *IRAP* con *NoRaslOutputFlag* 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 como sigue:

- 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 *de la imagen actual* 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 *con la misma ID de capa que la imagen actual*, NoOutputOfPriorPicsFlag se puede (pero no debe) establecer en 1 por 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 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 como sigue:

- Si NoOutputOfPriorPicsFlag es igual a 1, todos los búferes de almacenamiento de imágenes *que están en el DPB y que contienen imágenes que tienen el mismo valor de nuh\_layer\_id que el de la imagen actual*, se vacían sin emitir las imágenes que contienen, y la plenitud del DPB se *decrementa en el número de imágenes eliminadas*.

- De otro modo (NoOutputOfPriorPicsFlag es igual a 0), todos los búferes de almacenamiento de imágenes que contienen una imagen que está marcada como "no necesaria para la salida" y "no usada como referencia" se vacían (sin salida), y todos los búferes de almacenamiento de imágenes no vacíos en el DPB se vacían invocando repetidamente el proceso de "expulsión" especificado en la subcláusula F.13.5.2.4, y la plenitud del DPB se establece igual a 0.

- De otro modo (la imagen actual no es una imagen IRAP con NoRaslOutputFlag igual a 1 o con nuh\_layer\_id distinto de 0), se vacían todos los búferes de almacenamiento de imágenes que contienen una imagen marcada como "no necesaria para la salida" y "no usada como referencia" (sin salida). Para cada búfer de almacenamiento de imágenes que se vacía, la plenitud del DPB se decrementa 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 "expulsión" especificado en la subcláusula F.13.5.2.4 se invoca repetidamente mientras se decrementa aún más la plenitud del DPB en uno para cada búfer 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 el DPB que están marcadas como "necesarias para la salida" es mayor que sps\_max\_num\_reorder\_pics[HighestTid] del SPS activo (cuando currLayerId es igual a 0) o del 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 del SPS de la capa activa si el valor de currLayerId no es igual a 0 y hay al menos una imagen con nuh\_layer\_id igual a currLayerId en el DPB que está marcada como "necesaria para la salida" para la cual la variable asociada PicLatencyCount[currLayerId] es mayor o igual que SpsMaxLatencyPictures[HighestTid] derivada del SPS activo (cuando currLayerId es igual a 0) o del SPS de la capa activa para el valor de currLayerId.

- El número de imágenes con nuh\_layer\_id igual a currLayerId en el DPB es mayor o igual que sps\_max\_dec\_pic\_buffering\_minus1[HighestTid] + 1 del SPS activo (cuando currLayerId es igual a 0) o del SPS de la capa activa para el valor de currLayerId.

## **Implementación de ejemplo 2**

**[0193]** En los modos de realización descritos a continuación, no\_output\_of\_prior\_pics\_flag se señala solo para imágenes IRAP, y NoOutputOfPriorPicsFlag se deriva para todas las imágenes IRAP con NoRaslOutputFlag igual a un valor de "1" y se infiere, en base al valor de NoClRasOutputFlag, para todas las imágenes no IRAP que son imágenes LIP.

**Cambios en la eliminación de imágenes del DPB**

- 5 **[0194]** El siguiente texto de ejemplo se puede usar para implementar uno o más de los modos de realización descritos en el presente documento. Las adiciones y eliminaciones del lenguaje existente en la especificación de HEVC se muestran en *cursiva*.

**C.3.2 Eliminación de imágenes del DPB**

- 10 **[0195]** La eliminación de imágenes del DPB antes de la decodificación de la imagen actual (pero después de analizar sintácticamente la cabecera de fragmento del primer fragmento de la imagen actual) sucede instantáneamente en el momento de la eliminación del CPB de la primera unidad de decodificación de la unidad de acceso n (que contiene la imagen actual) y procede como sigue:
- 15 - Se invoca el proceso de decodificación para RPS como se especifica en la subcláusula 8.3.2.
- Si la imagen actual es *una LIP (o 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:
- 20     **1.** La variable NoOutputOfPriorPicsFlag se deriva para el decodificador bajo prueba como sigue:
- 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).
- 25 - 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 *de la imagen actual* es diferente del valor de pic\_width\_in\_luma\_samples, pic\_height\_in\_luma\_samples, o 1[HighestTid], respectivamente, derivado del SPS activo para la imagen anterior *con la misma ID de capa que la imagen actual*, NoOutputOfPriorPicsFlag se puede (pero no debe) establecer a 1 por el decodificador bajo prueba, independientemente del valor de no\_output\_of\_prior\_pics\_flag.
- 30     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.
- 35 - De otro modo, *si la imagen actual es una imagen IRAP*, NoOutputOfPriorPicsFlag se establece igual a no\_output\_of\_prior\_pics\_flag.
- *De otro modo, si la imagen actual (que es una LIP) no es una imagen IRAP y si NoClrasOutputFlag asociado a nuh\_layer\_id igual a 0 de la imagen IRAP en la unidad de acceso actual es igual a 1, NoOutputOfPriorPicsFlag se establece igual a 1.*
- 40 - *De otro modo, NoOutputOfPriorPicsFlag se establece igual a 0.*
- 45     **2.** El valor de NoOutputOfPriorPicsFlag derivado para el decodificador bajo prueba se aplica al HRD, de modo que cuando el valor de NoOutputOfPriorPicsFlag es igual a 1, todos los búferes de almacenamiento de imágenes *que están en el DPB y que contienen imágenes que tienen el mismo valor de nuh\_layer\_id que el de la imagen actual*, se vacían sin emitir las imágenes que contienen, y la plenitud del DPB se decrementa en el número de imágenes eliminadas.
- 50 - Cuando se cumplen las dos condiciones siguientes para cualquier imagen k en el DPB, todas esas imágenes k en el DPB se eliminan del DPB:
- la imagen k está marcada como "no usada como referencia"
- 55 - la imagen k tiene PicOutputFlag igual a 0 o su tiempo de salida del DPB es menor o igual que el tiempo de eliminación del 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)
- 60 - Para cada imagen que se elimina del DPB, la plenitud del DPB se decrementa en uno.

**Cambios en la salida y eliminación de imágenes del DPB**

- 65 **[0196]** El siguiente texto de ejemplo se puede usar para implementar uno o más de los modos de realización descritos en el presente documento. Las adiciones y eliminaciones del lenguaje existente en la especificación de

HEVC se muestran en *cursiva*.

#### F.13.5.2.2 Salida y eliminación de imágenes del DPB

- 5 **[0197]** La salida y eliminación de imágenes del DPB antes de descodificar la imagen actual (pero después de analizar sintácticamente la cabecera de fragmento del primer fragmento de la imagen actual) sucede instantáneamente cuando la primera unidad de descodificación de la imagen actual se elimina del CPB y procede de la siguiente manera:
- 10 Se invoca el proceso de descodificación para RPS como se especifica en la subcláusula F.8.3.2.
- Si la imagen actual es una *LIP* (o una imagen *IRAP* con *NoRaslOutputFlag* igual a 1 y con *nuh\_layer\_id* igual a 0) que no es la imagen 0, se aplican las siguientes etapas ordenadas:
 

15 1. La variable *NoOutputOfPriorPicsFlag* se deriva para el descodificador bajo prueba como sigue:

    - 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 de la imagen actual es diferente del valor de *pic\_width\_in\_luma\_samples*, *pic\_height\_in\_luma\_samples*, o *1*[*HighestTid*], respectivamente, derivado del SPS activo para la imagen anterior con la misma ID de capa que la imagen actual, *NoOutputOfPriorPicsFlag* se puede (pero no debe) establecer a 1 por el descodificador bajo prueba, independientemente del valor de *no\_output\_of\_prior\_pics\_flag*.

20 NOTA: aunque en estas condiciones se prefiere establecer *NoOutputOfPriorPicsFlag* igual a *no\_output\_of\_prior\_pics\_flag*, el descodificador bajo prueba puede establecer *NoOutputOfPriorPicsFlag* en 1 en este caso.

    - De otro modo, si la imagen actual es una imagen *IRAP*, *NoOutputOfPriorPicsFlag* se establece igual a *no\_output\_of\_prior\_pics\_flag*.
    - De otro modo, si la imagen actual (que es una *LIP*) no es una imagen *IRAP* y si *NoClrasOutputFlag* asociado a *nuh\_layer\_id* igual a 0 de la imagen *IRAP* en la unidad de acceso actual es igual a 1, *NoOutputOfPriorPicsFlag* se establece igual a 1.
    - De otro modo, *NoOutputOfPriorPicsFlag* se establece igual a 0.

25 2. El valor de *NoOutputOfPriorPicsFlag* derivado para el descodificador bajo prueba se aplica para el HRD como sigue:

    - Si *NoOutputOfPriorPicsFlag* es igual a 1, todos los búferes de almacenamiento de imágenes que están en el DPB y que contienen imágenes que tienen el *nuh\_layer\_id* igual al de la imagen actual, se vacían sin emitir las imágenes que contienen, y la totalidad del DPB se decrementa en el número de imágenes eliminadas.
    - De otro modo (*NoOutputOfPriorPicsFlag* es igual a 0), todos los búferes de almacenamiento de imágenes que contienen una imagen que está marcada como "no necesaria para la salida" y "no usada como referencia" se vacían (sin salida), y todos los búferes de almacenamiento de imágenes no vacíos en el DPB se vacían invocando repetidamente el proceso de "expulsión" especificado en la subcláusula F.13.5.2.4, y la plenitud del DPB se establece igual a 0.
    - De otro modo (la imagen actual no es una imagen *IRAP* con *NoRaslOutputFlag* igual a 1 o con *nuh\_layer\_id* distinto de 0), se vacían todos los búferes de almacenamiento de imágenes que contienen una imagen marcada como "no necesaria para la salida" y "no usada como referencia" (sin salida). Para cada búfer de almacenamiento de imágenes que se vacía, la plenitud del DPB se decrementa en uno. La variable *currLayerId* se establece igual a *nuh\_layer\_id* de la imagen descodificada actual y cuando una o más de las siguientes condiciones son verdaderas, el proceso de "expulsión" especificado en la subcláusula F.13.5.2.4 se invoca repetidamente mientras se decrementa aún más la plenitud del DPB en uno para cada búfer 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 el DPB que están marcadas como "necesarias para la salida" es mayor que *sps\_max\_num\_reorder\_pics*[*HighestTid*] del SPS activo (cuando *currLayerId* es igual a 0) o del SPS de la capa activa para el valor de *currLayerId* (cuando

30

35

40

45

50

55

60

65

currLayerId no es igual a 0).

- sps\_max\_latency\_increase\_plus1[HighestTid] del SPS activo (cuando currLayerId es igual a 0) o del SPS de la capa activa si el valor de currLayerId no es igual a 0 y hay al menos una imagen con nuh\_layer\_id igual a currLayerId en el DPB que está marcada como "necesaria para la salida" para la cual la variable asociada PicLatencyCount[currLayerId] es mayor o igual que SpsMaxLatencyPictures[HighestTid] derivada del SPS activo (cuando currLayerId es igual a 0) o del SPS de la capa activa para el valor de currLayerId.
- El número de imágenes con nuh\_layer\_id igual a currLayerId en el DPB es mayor o igual que sps\_max\_dec\_pic\_buffering\_minus1[HighestTid] + 1 del SPS activo (cuando currLayerId es igual a 0) o del SPS de la capa activa para el valor de currLayerId.

**[0198]** En algunos modos de realización, tanto de conformidad de temporización de salida como de conformidad de orden de salida, cuando la imagen actual es una imagen LIP que no es una imagen IRAP, y cuando NoClasOutputFlag está asociado a otra imagen que es una imagen IRAP, con nuh\_layer\_id igual a un valor de "0", en la unidad de acceso actual es igual a un valor de "1", NoOutputOfPriorPicsFlag se establece igual a NoOutputOfPriorPicsFlag de la imagen IRAP en la AU actual que tiene nuh\_layer\_id igual a un valor de "0".

### **Implementación de ejemplo 3**

**[0199]** En los modos de realización descritos a continuación, se especifica que la descarga de imágenes en el momento de descodificar imágenes IRAP no BL con NoRasOutputFlag igual a un valor de "1" se realiza de una manera específica de la capa. Cuando la descarga se realiza en el momento de descodificar imágenes BL IRAP con NoRasOutputFlag igual a un valor de "1", se especifica que la operación de descarga se debe realizar en todas las capas. Por ejemplo, toda la descarga es específica de la capa para las capas no base (por ejemplo, capas de mejora), pero una operación de descarga realizada en relación con la capa base puede descargar las imágenes en las capas no base.

### **Cambios en la eliminación de imágenes del DPB**

**[0200]** El siguiente texto de ejemplo se puede usar para implementar uno o más de los modos de realización descritos en el presente documento. Las adiciones y eliminaciones del lenguaje existente en la especificación de HEVC se muestran en *cursiva*.

#### **C.3.2 Eliminación de imágenes del DPB**

**[0201]** La eliminación de imágenes del DPB antes de la descodificación de la imagen actual (pero después de analizar sintácticamente la cabecera de fragmento del primer fragmento de la imagen actual) sucede instantáneamente en el momento de la eliminación del CPB de la primera unidad de descodificación de la unidad de acceso n (que contiene la imagen actual) y procede como sigue:

- Se invoca el proceso de descodificación para RPS como se especifica en la subcláusula 8.3.2.
- Cuando 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, o una imagen IRAP con NoRasOutputFlag igual a 1 con nuh\_layer\_id distinto de 0 y que no pertenece a una IRAP AU inicial*, se aplican las siguientes etapas ordenadas:

1. La variable NoOutputOfPriorPicsFlag se deriva para el descodificador bajo prueba como sigue:

- Si la imagen actual es una imagen CRA *con nuh\_layer\_id igual a 0*, 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 *de la imagen actual* 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 *con la misma ID de capa que la imagen actual*, NoOutputOfPriorPicsFlag se puede (pero no debe) establecer en 1 por el descodificador 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 descodificador bajo prueba puede establecer NoOutputOfPriorPicsFlag en 1 en este caso.

- De otro modo, NoOutputOfPriorPicsFlag se establece igual a no\_output\_of\_prior\_pics\_flag.

1. El valor de NoOutputOfPriorPicsFlag derivado para el decodificador bajo prueba se aplica para el HRD como sigue:

- si el valor de NoOutputOfPriorPicsFlag es igual a 1 y *nuh\_layer\_id* de la imagen actual es igual a 0, todos los búferes de almacenamiento de imágenes en el DPB se vacían sin emitir las imágenes que contienen, y la plenitud del DPB se establece igual a 0.
- De otro modo, si NoOutputOfPriorPicsFlag es igual a 1 y *nuh\_layer\_id* de la imagen actual es distinto de 0, todos los búferes de almacenamiento de imágenes que están en el DPB y que contienen imágenes que tienen *nuh\_layer\_id* igual al de la imagen actual, se vacían y la plenitud del DPB se decrementa en el número de imágenes eliminadas.
- Cuando se cumplen las dos condiciones siguientes para cualquier imagen k en el DPB, todas esas imágenes k en el DPB se eliminan del DPB:
  - la imagen k está marcada como "no usada como referencia"
  - la imagen k tiene PicOutputFlag igual a 0 o su tiempo de salida del DPB es menor o igual que el tiempo de eliminación del 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 del DPB, la plenitud del DPB se decrementa en uno.

#### **Cambios en la salida y eliminación de imágenes del DPB**

**[0202]** El siguiente texto de ejemplo se puede usar para implementar uno o más de los modos de realización descritos en el presente documento. Las adiciones y eliminaciones del lenguaje existente en la especificación de HEVC se muestran en *cursiva*.

##### **F.13.5.2.2 Salida y eliminación de imágenes del DPB**

**[0203]** La salida y eliminación de imágenes del DPB antes de decodificar la imagen actual (pero después de analizar sintácticamente la cabecera de fragmento del primer fragmento de la imagen actual) sucede instantáneamente cuando la primera unidad de decodificación de la imagen actual se elimina del CPB y procede de la siguiente manera:

Se invoca el proceso de decodificación para RPS como se especifica en la subcláusula F.8.3.2.

- Si la imagen actual es una imagen IRAP con NoRaslOutputFlag igual a 1 y con *nuh\_layer\_id* igual a 0 que no es la imagen 0, o una imagen IRAP con NoRaslOutputFlag igual a 1 con *nuh\_layer\_id* distinto de 0 y que no pertenece a una IRAP AU inicial, se aplican las siguientes etapas ordenadas:

1. La variable NoOutputOfPriorPicsFlag se deriva para el decodificador bajo prueba como sigue:

- Si la imagen actual es una imagen CRA con *nuh\_layer\_id* igual a 0, 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 de la imagen actual es diferente del valor de pic\_width\_in\_luma\_samples, pic\_height\_in\_luma\_samples, o 1[HighestTid], respectivamente, derivado del SPS activo para la imagen anterior con la misma ID de capa que la imagen actual, NoOutputOfPriorPicsFlag se puede (pero no debe) establecer a 1 por 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 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 como sigue:

- Si NoOutputOfPriorPicsFlag es igual a 1 y *nuh\_layer\_id* de la imagen actual es igual a 0, todos los búferes de almacenamiento de imágenes en el DPB se vacían sin emitir las imágenes que contienen,

y la plenitud del DPB se establece igual a 0.

- De otro modo, si *NoOutputOfPriorPicsFlag* es igual a 1 y *nuh\_layer\_id* de la imagen actual es distinto de 0, todos los búferes de almacenamiento de imágenes que están en el DPB y que contienen imágenes que tienen *nuh\_layer\_id* igual al de la imagen actual, se vacían y la plenitud del DPB se decrementa en el número de imágenes eliminadas.

- De otro modo (*NoOutputOfPriorPicsFlag* es igual a 0), todos los búferes de almacenamiento de imágenes que contienen una imagen que está marcada como "no necesaria para la salida" y "no usada como referencia" se vacían (sin salida), y todos los búferes de almacenamiento de imágenes no vacíos en el DPB se vacían invocando repetidamente el proceso de "expulsión" especificado en la subcláusula F.13.5.2.4, y la plenitud del DPB se establece igual a 0.

- De otro modo (la imagen actual no es una imagen IRAP con *NoRaslOutputFlag* igual a 1 o con *nuh\_layer\_id* distinto de 0), se vacían todos los búferes de almacenamiento de imágenes que contienen una imagen marcada como "no necesaria para la salida" y "no usada como referencia" (sin salida). Para cada búfer de almacenamiento de imágenes que se vacía, la plenitud del DPB se decrementa en uno. La variable *currLayerId* se establece igual a *nuh\_layer\_id* de la imagen descodificada actual y cuando una o más de las siguientes condiciones son verdaderas, el proceso de "expulsión" especificado en la subcláusula F.13.5.2.4 se invoca repetidamente mientras se decrementa aún más la plenitud del DPB en uno para cada búfer 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 el DPB que están marcadas como "necesarias para la salida" es mayor que *sps\_max\_num\_reorder\_pics*[*HighestTid*] del SPS activo (cuando *currLayerId* es igual a 0) o del 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 del SPS de la capa activa si el valor de *currLayerId* no es igual a 0 y hay al menos una imagen con *nuh\_layer\_id* igual a *currLayerId* en el DPB que está marcada como "necesaria para la salida" para la cual la variable asociada *PicLatencyCount*[*currLayerId*] es mayor o igual que *SpsMaxLatencyPictures*[*HighestTid*] derivada del SPS activo (cuando *currLayerId* es igual a 0) o del SPS de la capa activa para el valor de *currLayerId*.

- El número de imágenes con *nuh\_layer\_id* igual a *currLayerId* en el DPB es mayor o igual que *sps\_max\_dec\_pic\_buffering\_minus1*[*HighestTid*] + 1 del SPS activo (cuando *currLayerId* es igual a 0) o del SPS de la capa activa para el valor de *currLayerId*.

#### **Otras Consideraciones**

**[0204]** La información y las señales divulgadas en el presente documento se pueden representar usando cualquiera de una variedad de tecnologías y técnicas diferentes. Por ejemplo, los datos, las instrucciones, los comandos, la información, las señales, los bits, los símbolos y los chips que se pueden haber mencionado a lo largo de la descripción anterior se pueden representar mediante tensiones, corrientes, ondas electromagnéticas, campos o partículas magnéticos, campos o partículas ópticos, o cualquier combinación de los mismos.

**[0205]** Los diversos bloques lógicos, módulos, circuitos y etapas de algoritmo ilustrativos descritos en relación con los modos de realización divulgados en el presente documento se pueden implementar como hardware electrónico, software informático o combinaciones de ambos. Para ilustrar claramente esta intercambiabilidad de hardware y software, anteriormente se han descrito en general diversos componentes, bloques, módulos, circuitos y etapas ilustrativos en términos de su funcionalidad. Que dicha funcionalidad se implemente como hardware o software depende de las restricciones de la aplicación y del diseño en particular impuestas al sistema global. Los expertos en la técnica pueden implementar la funcionalidad descrita de formas distintas para cada aplicación particular, pero no debería interpretarse que dichas decisiones de implementación suponen apartarse del alcance de la presente invención.

**[0206]** Las técnicas descritas en el presente documento se pueden implementar en hardware, software, firmware o en cualquier combinación de los mismos. Dichas técnicas se pueden implementar 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. Todos los rasgos característicos descritos como módulos o componentes se pueden implementar juntos en un dispositivo lógico integrado o por separado, como dispositivos lógicos discretos pero interoperables. Si se implementan en software, las técnicas se pueden realizar, al menos en parte, por un medio de almacenamiento de datos legible por ordenador que comprenda código de programa que incluya instrucciones que, cuando se ejecuten, realicen 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 borrrable 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 al que se puede acceder, leer y/o ejecutar por medio de un ordenador, tales como señales u ondas propagadas.

**[0207]** El código de programa se puede ejecutar 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 la presente 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 se puede implementar 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, se puede referir a cualquier estructura anterior, cualquier combinación de la estructura anterior, o cualquier otra estructura o aparato adecuados para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento se puede proporcionar 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.

**[0208]** Las técnicas de la presente 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 la presente 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 mediante diferentes unidades de hardware. En su lugar, como se describe anteriormente, diversas unidades se pueden combinar en una unidad de hardware de códec o proporcionar mediante un grupo de unidades de hardware interoperativas, que incluya uno o más procesadores como se describe anteriormente, junto con software y/o firmware adecuados.

**[0209]** Se han descrito diversos modos de realización de la invención. Estos y otros modos de realización están dentro del alcance de las siguientes reivindicaciones.

## REIVINDICACIONES

1. Un aparato configurado para codificar información de vídeo de acuerdo con SHVC y/o MV-HEVC, comprendiendo el aparato:

una unidad de memoria configurada para almacenar información de vídeo asociada con una capa base (BL) y una capa de mejora (EL), teniendo la BL una imagen de BL (422B) en una primera unidad de acceso, y teniendo la EL una imagen de EL (422A) en la primera unidad de acceso, en el que la imagen de BL es una imagen de punto de acceso aleatorio interno (IRAP) y la imagen de EL es una imagen de inicialización de capa (LIP) que no es una imagen IRAP, y en el que la imagen de BL tiene un NoCrasOutputFlag asociado a ella y en el que la imagen de BL se asocia a una ID de capa más pequeña de todas las ID de capa usadas para la información de vídeo; y

un procesador en comunicación con la unidad de memoria, estando configurado el procesador para:

determinar si un valor de NoCrasOutputFlag es igual a un primer valor, en el que NoCrasOutputFlag es indicativo de si se va a emitir una imagen de omisión de acceso aleatorio de capas cruzadas (CL-RAS), en el que una imagen CL-RAS es una imagen de EL que es no es una imagen IRAP y está en la misma unidad de acceso que una imagen BL IRAP;

establecer un valor de NoOutputOfPriorPicsFlag asociado a la imagen de EL igual a un segundo valor si el NoCrasOutputFlag es igual al primer valor; y

eliminar, cuando el valor de NoOutputOfPriorPicsFlag es igual al segundo valor, una o más imágenes de EL en un búfer de imágenes descodificadas (DPB) sin emitir la una o más imágenes de EL antes de que se codifique la imagen de EL.

2. El aparato de la reivindicación 1, en el que el NoCrasOutputFlag asociado con la imagen de BL indica además si la primera unidad de acceso sigue inmediatamente a un punto de empalme donde dos flujos de bits se unen entre sí en un único flujo de bits que comprende la BL y la EL.

3. El aparato de la reivindicación 1, en el que el procesador está configurado además para eliminar, en base a una determinación de que la imagen de EL no es una imagen IRAP y de que el valor de NoCrasOutputFlag es igual a 1, la una o más imágenes de EL en el DPB sin emitir la una o más imágenes de EL antes de que se codifique la imagen de EL.

4. El aparato de la reivindicación 1, en el que la primera unidad de acceso es una unidad de acceso IRAP inicial.

5. El aparato de la reivindicación 1, en el que la imagen de BL tiene NoRasOutputFlag igual a 1.

6. Un procedimiento de codificación de información de vídeo de acuerdo con SHVC y/o MV-HEVC, estando asociada la información de vídeo a una capa base (BL) y una capa de mejora (EL), teniendo la BL una imagen de BL (422B) en una primera unidad de acceso, y teniendo la EL una imagen de EL (422A) en la primera unidad de acceso, en el que la imagen de BL es una imagen de punto de acceso aleatorio interno (IRAP) y la imagen de EL es una imagen de inicialización de capa (LIP) que no es una imagen IRAP, y en el que la imagen de BL tiene un NoCrasOutputFlag asociada a ella y en el que la imagen de BL se asocia a la ID de capa más pequeña de todas las ID de capa usadas para la información de vídeo, comprendiendo el procedimiento:

determinar si un valor del NoCrasOutputFlag es igual a un primer valor, en el que NoCrasOutputFlag es indicativo de si se va a emitir una imagen de omisión de acceso aleatorio de capas cruzadas (CL-RAS), en el que una imagen CL-RAS es una imagen de EL que es no es una imagen IRAP y está en la misma unidad de acceso que una imagen BL IRAP;

establecer un valor de NoOutputOfPriorPicsFlag asociado a la imagen de EL igual a un segundo valor si el NoCrasOutputFlag es igual al primer valor; y

eliminar, cuando el valor de NoOutputOfPriorPicsFlag es igual al segundo valor, una o más imágenes de la capa de mejora (EL) en un búfer de imágenes descodificadas (DPB) sin emitir la una o más imágenes de EL antes de que se codifique la imagen de EL en la primera unidad de acceso.

7. El procedimiento de la reivindicación 6, en el que el NoCrasOutputFlag asociado con la imagen de BL indica si la primera unidad de acceso sigue inmediatamente a un punto de empalme donde dos flujos de bits se unen entre sí en un único flujo de bits que comprende la BL y la EL.

8. El procedimiento de la reivindicación 6, que comprende además al menos uno de (1) eliminar, en base a determinar que la imagen de EL no es una imagen IRAP y que el valor de NoCrasOutputFlag es igual a 1, la

una o más imágenes de EL en el DPB sin emitir la una o más imágenes de EL antes de que se codifique la imagen de EL.

- 5      **9.** Un medio no transitorio legible por ordenador que comprende código que, cuando se ejecuta, hace que un aparato realice el procedimiento de las reivindicaciones 6 a 8.

10

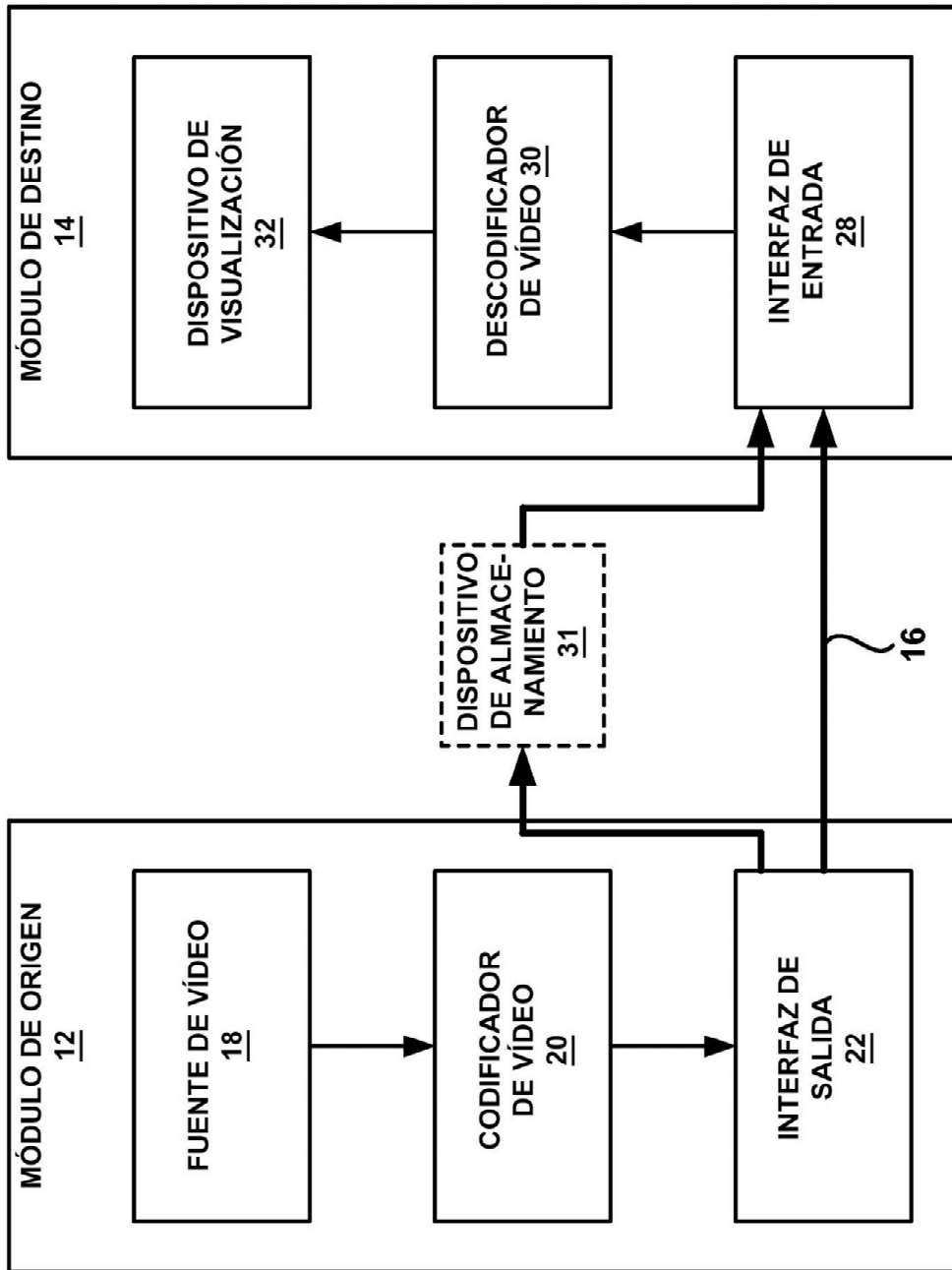


FIG. 1A

10'

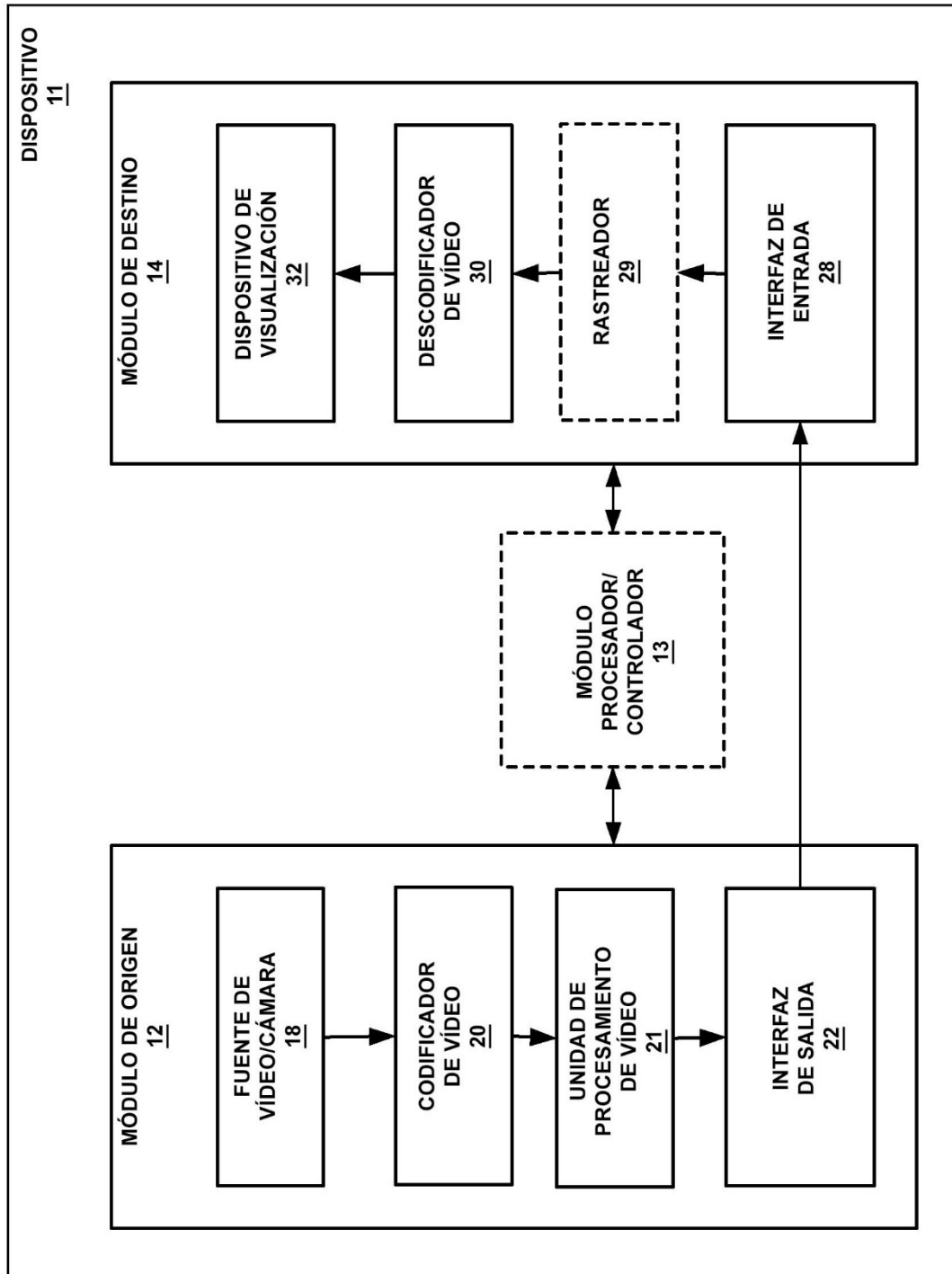


FIG. 1B

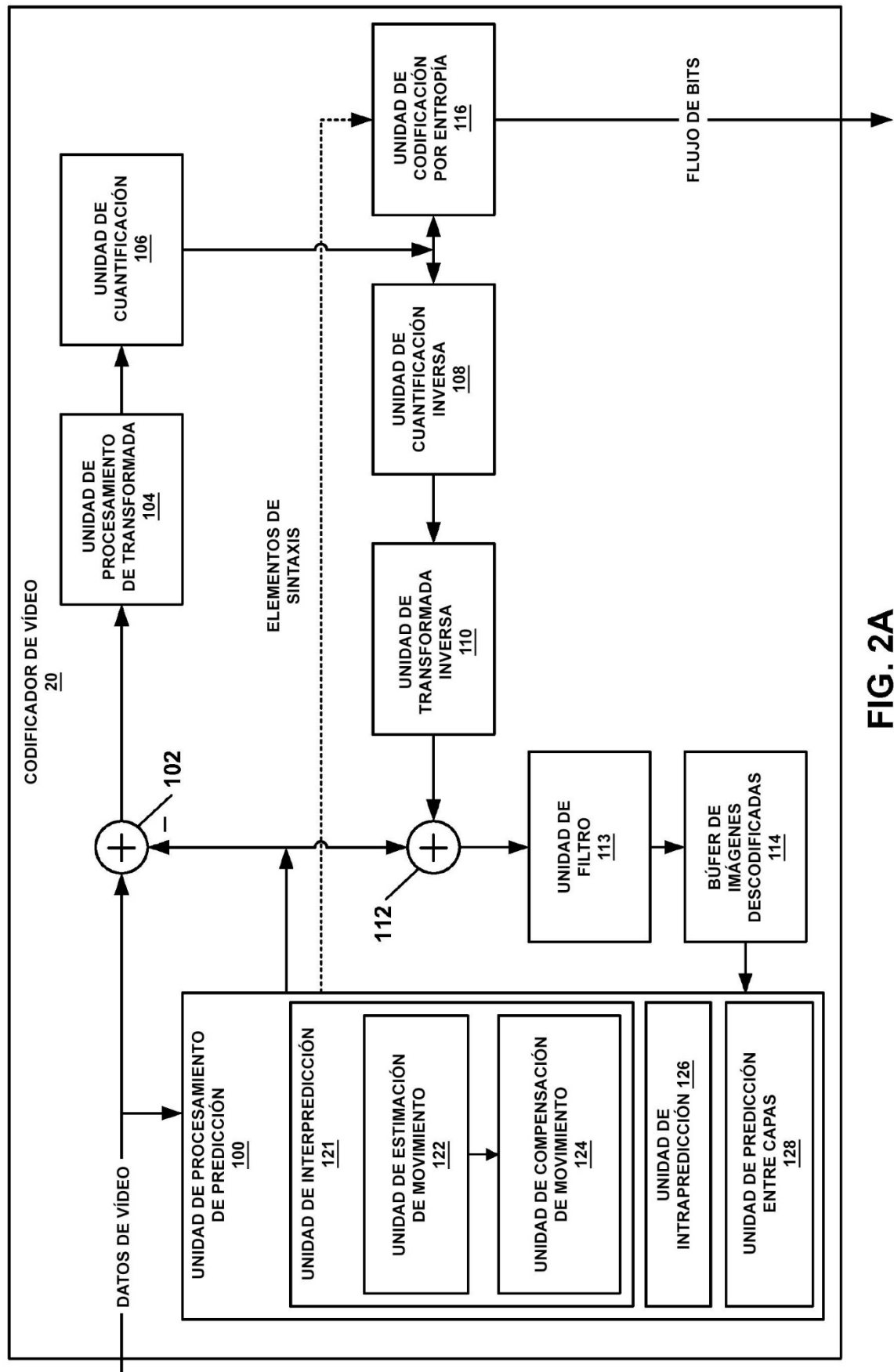


FIG. 2A

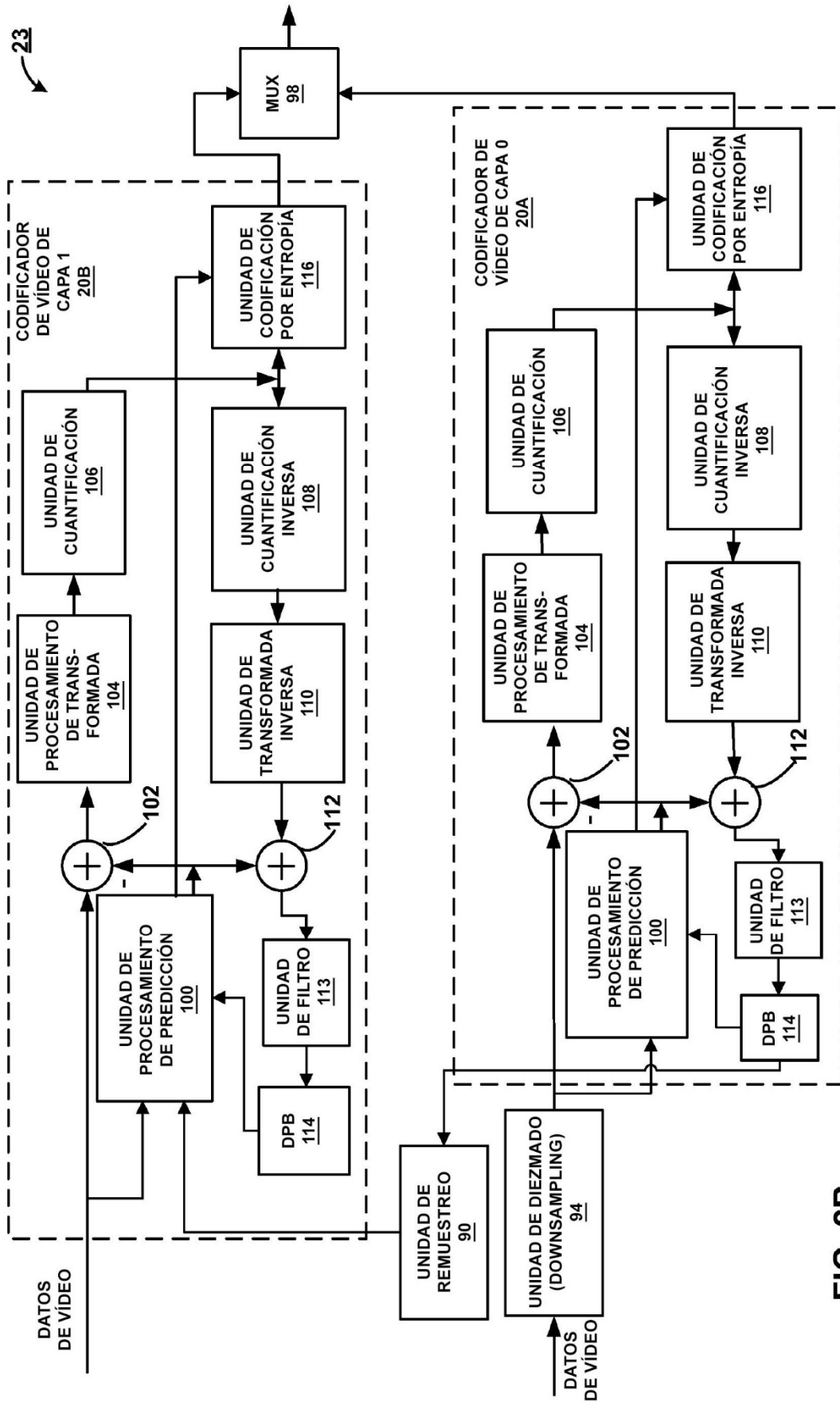


FIG. 2B

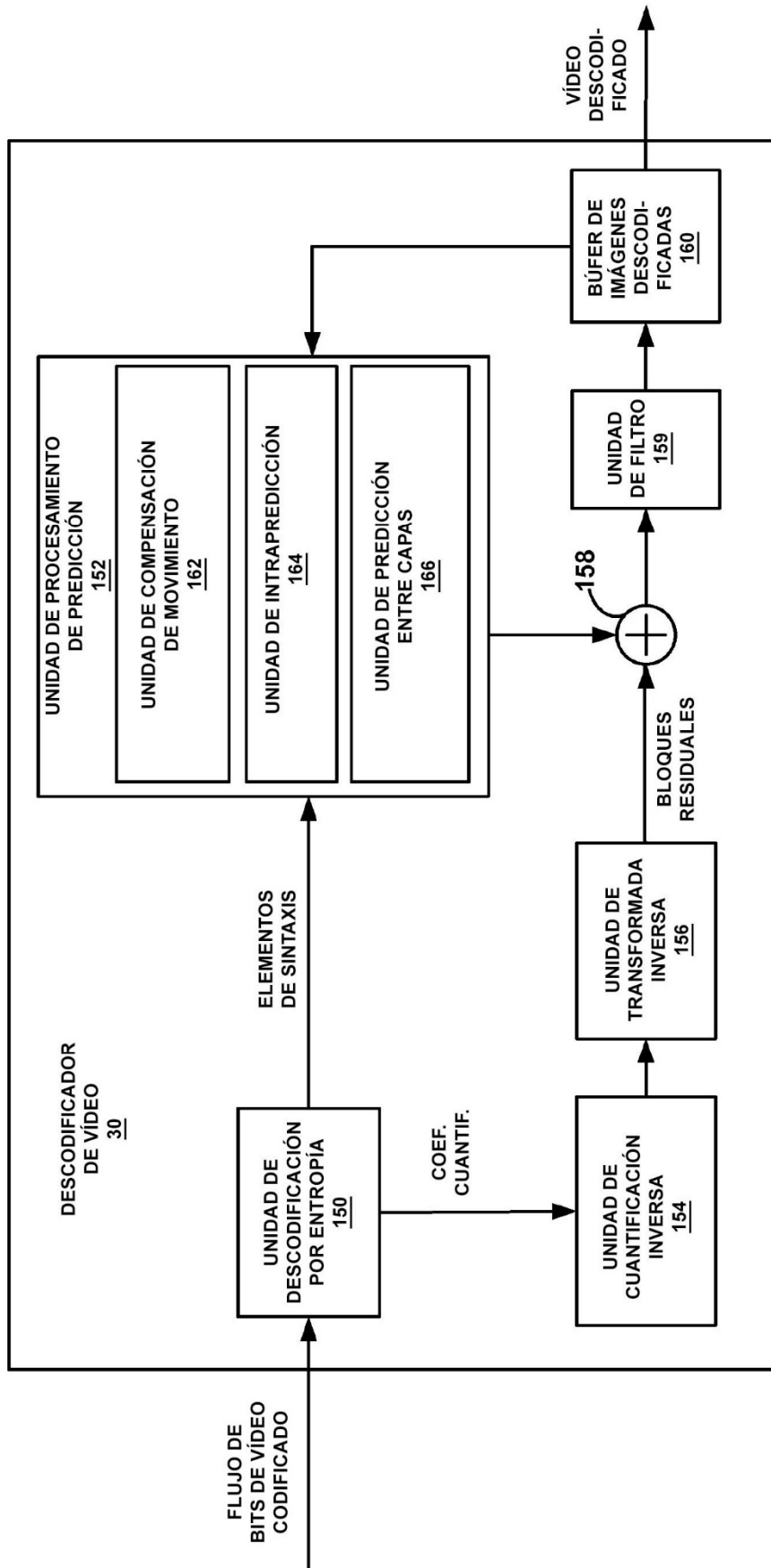
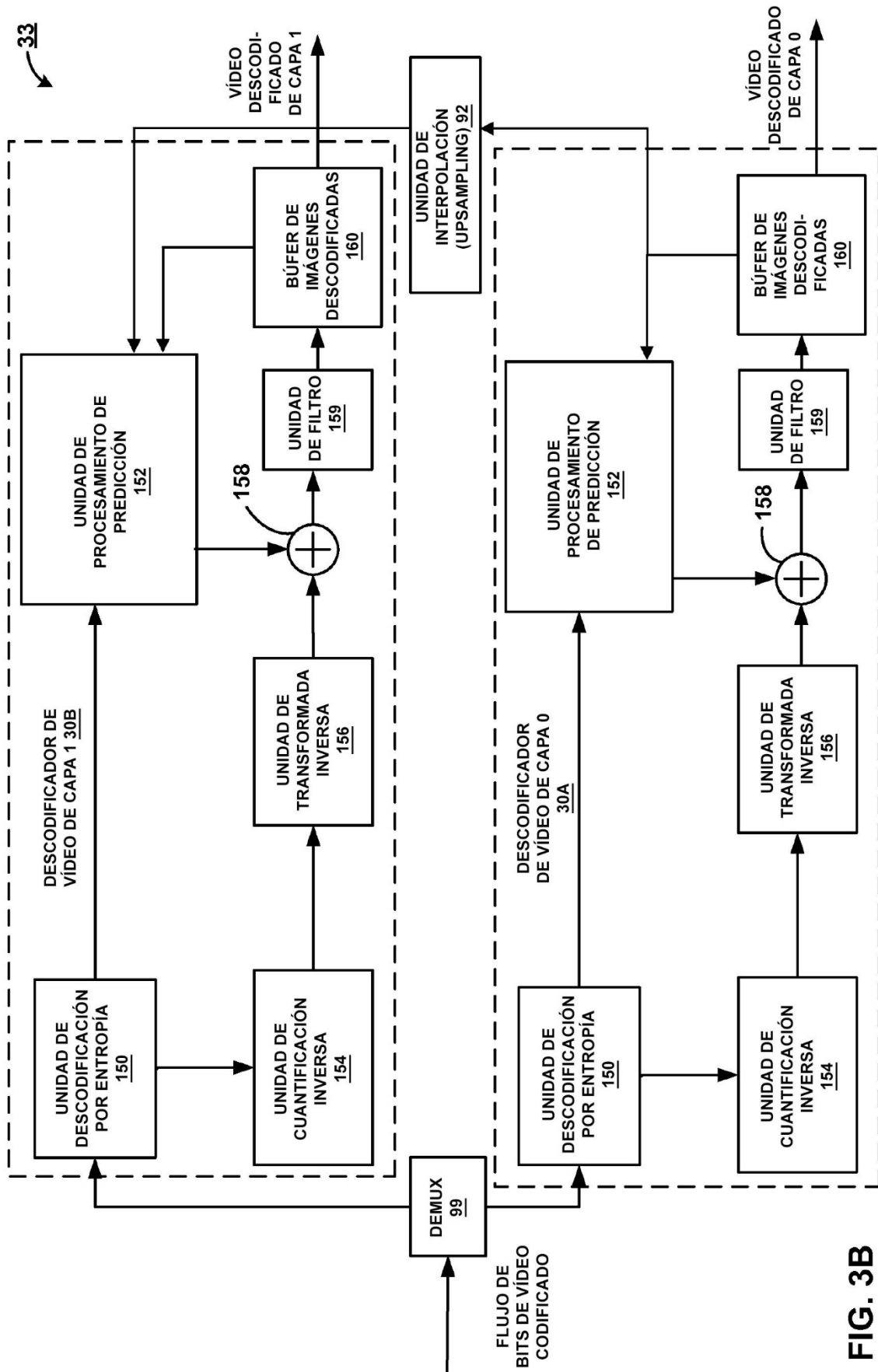
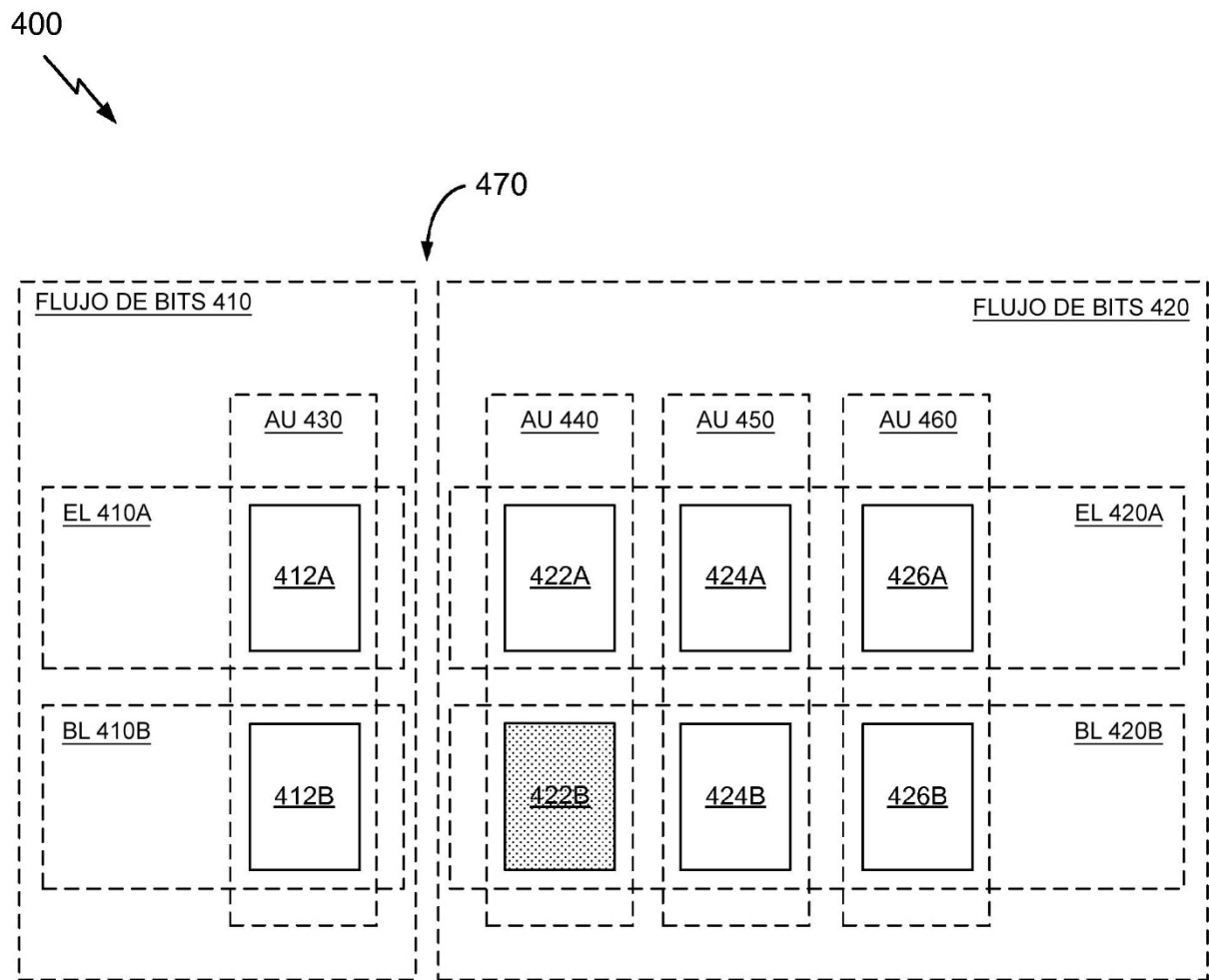
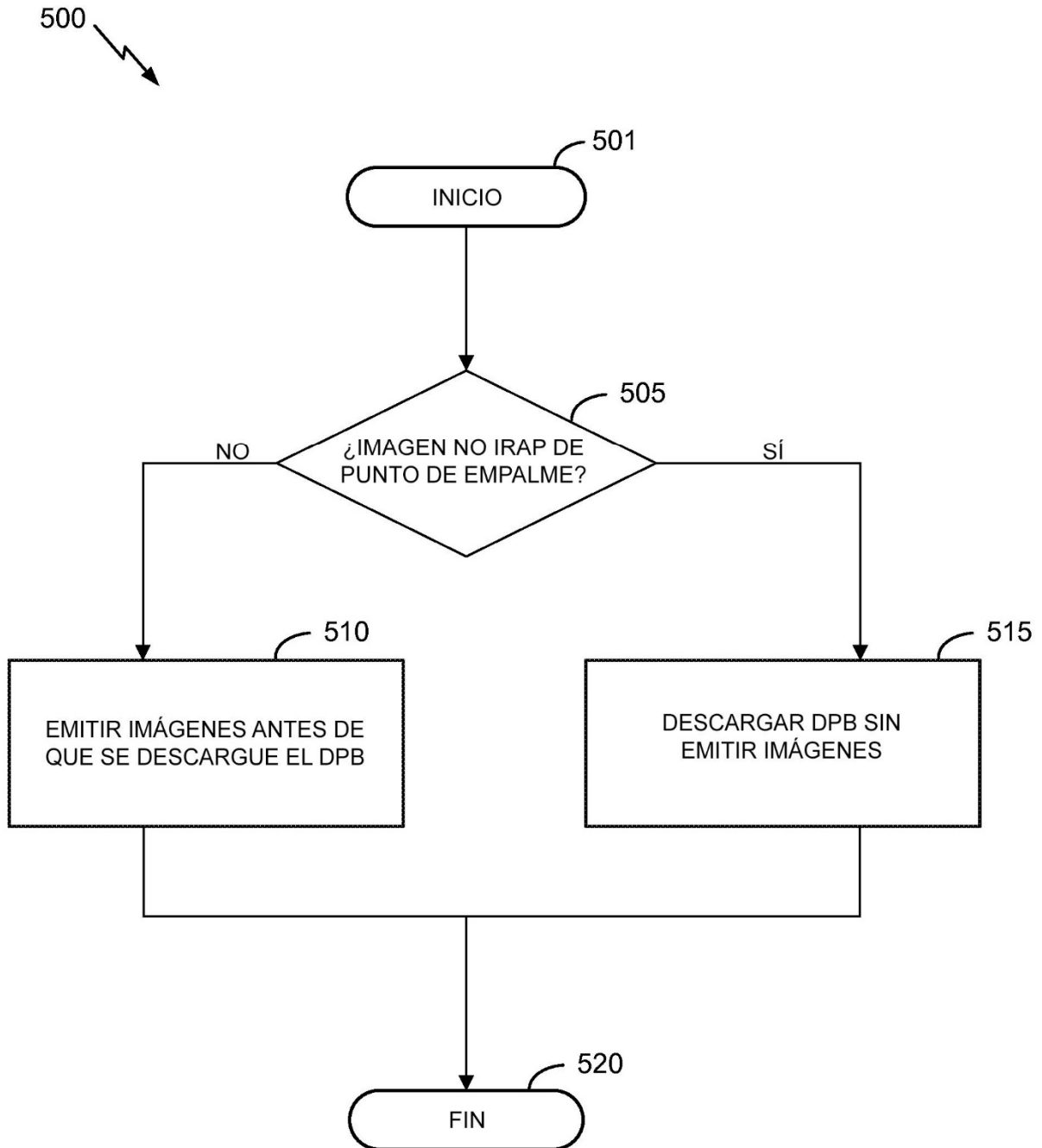


FIG. 3A

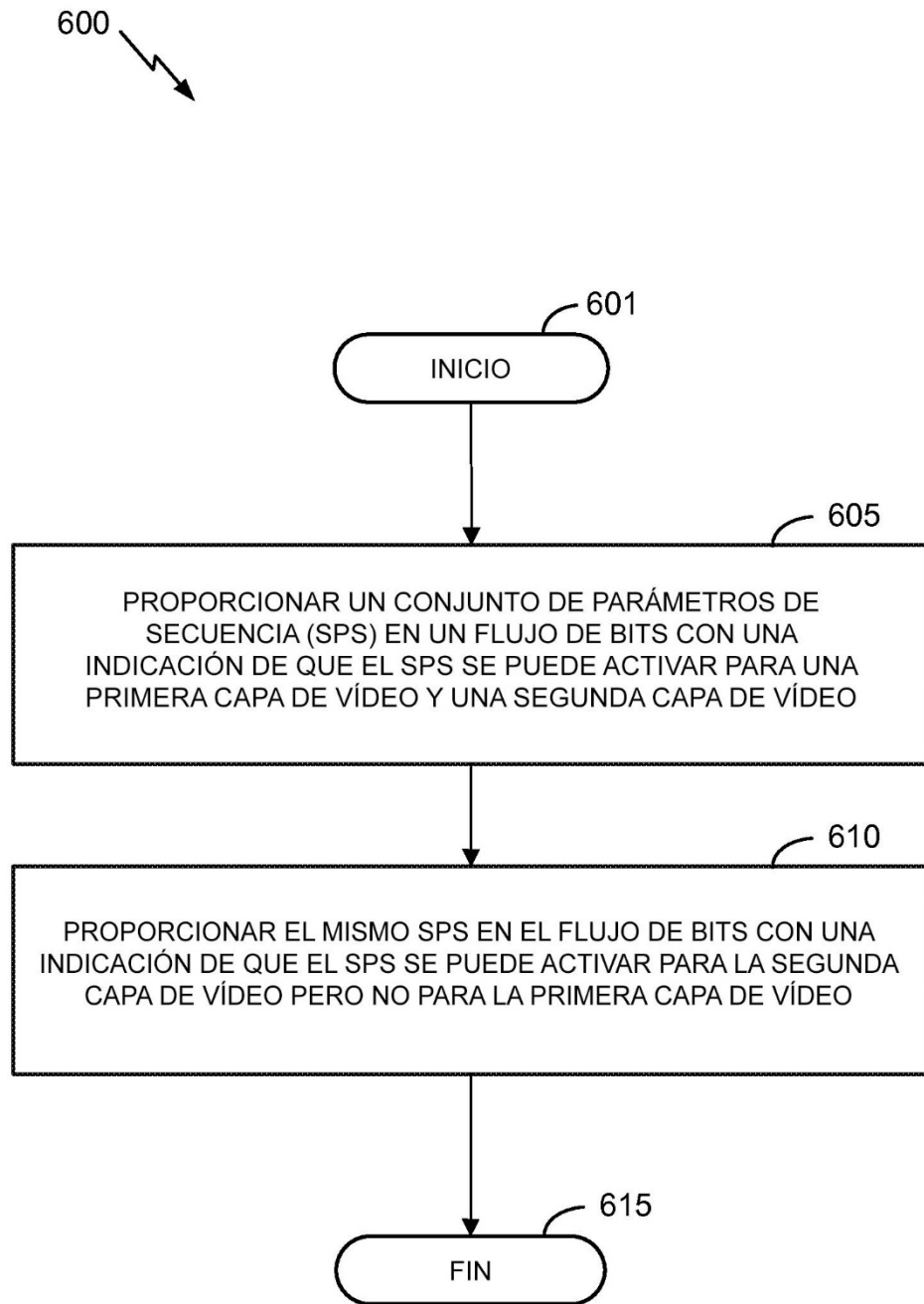




**FIG. 4**



**FIG. 5**

**FIG. 6**