

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 744 201**

51 Int. Cl.:

H04N 19/70 (2014.01)

H04N 19/46 (2014.01)

H04N 19/30 (2014.01)

H04N 19/68 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **14.10.2014 PCT/US2014/060492**

87 Fecha y número de publicación internacional: **23.04.2015 WO15057711**

96 Fecha de presentación y número de la solicitud europea: **14.10.2014 E 14792943 (4)**

97 Fecha y número de publicación de la concesión europea: **05.06.2019 EP 3058744**

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

30 Prioridad:

14.10.2013 US 201361890868 P
13.10.2014 US 201414513030

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
24.02.2020

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;
CHEN, YING;
WANG, YE-KUI y
HENDRY, FNU

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 744 201 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 la codificación escalable de información de vídeo

5 **CAMPO TÉCNICO**

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

10

ANTECEDENTES

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

15

20

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

25

30

[0004] La predicción espacial o temporal da como resultado un bloque predictivo para un bloque a codificar. Los datos residuales representan diferencias de píxeles entre el bloque original a codificar y el bloque predictivo. Un bloque intercodificado se codifica de acuerdo con un vector de movimiento que apunta a un bloque de muestras de referencia que forman el bloque predictivo, y los datos residuales que indican la diferencia entre el bloque codificado y el bloque predictivo. Un bloque intracodificado se codifica de acuerdo con un modo de intracodificación y los datos residuales. Para una mayor compresión, los datos residuales pueden transformarse desde el dominio del píxel a un dominio de transformación, dando como resultado coeficientes de transformación residuales, que a continuación se pueden cuantificar. Los coeficientes de transformación cuantificados, dispuestos inicialmente en una matriz bidimensional, pueden escanearse con el fin de generar un vector unidimensional de coeficientes de transformación, y puede aplicarse codificación por entropía para lograr aún más compresión.

35

40

45

SUMARIO

[0005] La codificación de vídeo escalable (SVC) se refiere a la codificación de vídeo en la que se utilizan una capa base (BL), a veces denominada capa de referencia (RL), y una o más capas de mejora escalable (EL). Para SVC, la capa base puede transportar datos de vídeo con un nivel de calidad base. Las una o más capas de mejora pueden transportar datos de vídeo adicionales para admitir, por ejemplo, niveles espaciales, temporales y/o de señal a ruido (SNR) más altos. Las capas de mejora pueden definirse en relación con una capa previamente codificada. Por ejemplo, una capa inferior puede servir como una BL, mientras que una capa superior puede servir como una EL. Las capas intermedias pueden servir como EL o RL, o como ambas. Por ejemplo, una capa intermedia (por ejemplo, una capa que ni es la capa más baja ni la capa más alta) puede ser una EL para las capas debajo de la capa intermedia, como la capa base o cualquier capa de mejora intermedia, y al mismo tiempo servir como RL para una o más capas de mejora sobre la capa intermedia. De manera similar, en la ampliación de múltiples vistas o 3D de la norma HEVC, puede haber múltiples vistas, y la información de una vista puede utilizarse para codificar (por ejemplo, codificar o descodificar) la información de otra vista (por ejemplo, estimación de movimiento, predicción del vector de movimiento y/u otras redundancias).

50

55

60

[0006] En SVC, se puede usar un recuento de orden de imágenes (POC) para indicar el orden en el que se deben emitir o mostrar las imágenes. Además, en algunas implementaciones, el valor del POC puede restablecerse (por ejemplo, establecerse en cero, establecerse en algún valor señalado en el flujo de bits u obtenerse de la información incluida en el flujo de bits) siempre que ciertos tipos de imágenes aparezcan en el flujo de bits. Por ejemplo, cuando ciertas imágenes de punto de acceso aleatorio aparecen en el flujo de bits, el POC puede

65

restablecerse. Cuando se restablece el POC de una imagen en particular, los POC de cualquier imagen que preceda a la imagen en particular en el orden de descodificación también se pueden restablecer, por ejemplo, para mantener el orden relativo en que esas imágenes se deben emitir o mostrar. Los POC de cualquier imagen que sigue a la imagen en particular en el orden de descodificación pueden señalizarse en el flujo de bits, suponiendo que el restablecimiento de POC se realizó en relación con la imagen en particular. Por ejemplo, si el POC se restablece a un valor de 0 en la Imagen A que precede inmediatamente a la Imagen B en orden de descodificación y orden de salida, el POC señalado en el flujo de bits para la Imagen B puede tener un valor de 1.

[0007] Sin embargo, en ciertos casos, la imagen en particular puede no estar disponible para el descodificador. Por ejemplo, la imagen en particular puede perderse durante la transmisión o puede eliminarse del flujo de bits para satisfacer las restricciones de ancho de banda. En tal caso, es posible que el descodificador no sepa restablecer los POC de las imágenes que preceden a la imagen en particular en el orden de descodificación. Esto es problemático porque los POC de las imágenes que siguen a la imagen en particular en el orden de descodificación se señalizan u obtienen como si el restablecimiento del POC se hubiera realizado en la imagen en particular. Por lo tanto, en tal caso, el orden relativo entre las imágenes que preceden a la imagen en particular y las imágenes que siguen a la imagen en particular puede volverse incorrecto. BOYCE J: "BoG report on random access and cross-layer alignment of pictures types [Informe BoG sobre el acceso aleatorio y la alineación de capas cruzadas de tipos de imágenes]", JCTVC-N0373, 14. CONFERENCIA JCT-VC; 25-7-2013 - 2-8-2013, se propone abordar los problemas que surgen de la alineación del POC cuando los IRAP no están alineados en las unidades de acceso (AU).

[0008] Por lo tanto, se desea un procedimiento de codificación mejorado para obtener los valores de POC, especialmente en el caso de que ciertas imágenes no estén disponibles. La invención a este efecto se establece en las reivindicaciones adjuntas al presente documento.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

[0009]

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

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

La **figura 2A** es un diagrama de bloques que ilustra un ejemplo de un codificador de vídeo que puede implementar técnicas según aspectos descritos en esta divulgación.

La **figura 2B** es un diagrama de bloques que ilustra un ejemplo de un codificador de vídeo que puede implementar técnicas según aspectos descritos en esta divulgación.

La **figura 3A** es un diagrama de bloques que ilustra un ejemplo de un descodificador de vídeo que puede implementar técnicas según aspectos descritos en esta divulgación.

La **figura 3B** es un diagrama de bloques que ilustra un ejemplo de un descodificador de vídeo que puede implementar técnicas según aspectos descritos en esta divulgación.

La **figura 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 **figura 5** es una tabla que ilustra los valores de POC de imágenes en diferentes capas, de acuerdo con un modo de realización de la presente divulgación.

La **figura 6** 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 **figura 7** es una tabla que ilustra los valores de POC de imágenes en diferentes capas, de acuerdo con un modo de realización de la presente divulgación.

La **figura 8** 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 **figura 9** 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 **figura 10** 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.

5 La **figura 11** 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 **figura 12** 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.

10 DESCRIPCIÓN DETALLADA

15 **[0010]** Ciertos 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, como HEVC (codificación de vídeo de alta eficiencia). Más específicamente, la presente divulgación se refiere a sistemas y procedimientos para mejorar el rendimiento de la predicción entre capas en la ampliación de codificación de vídeo escalable (SVC) de HEVC.

20 **[0011]** En la descripción a continuación, se describen técnicas H.264/AVC relacionadas con ciertos modos de realización; también se analizan la norma HEVC y las técnicas relacionadas. Aunque ciertos modos de realización se describen en el presente documento en el contexto de las normas HEVC y/o H.264, un experto en la materia puede apreciar que los sistemas y procedimientos divulgados en el presente documento pueden aplicarse a cualquier norma de codificación de vídeo adecuada. Por ejemplo, los modos de realización divulgados en el presente documento pueden ser aplicables a una o más de las siguientes normas: ITU-T H.261, MPEG-1 Visual de la ISO/IEC, ITU-T H.262 o MPEG-2 Visual de la ISO/IEC, ITU-T H.263, MPEG-4 Visual de la ISO/IEC e ITU-T H.264 (también conocida como AVC del MPEG-4 de la ISO/IEC), incluyendo sus ampliaciones de codificación de vídeo escalable (SVC) y de codificación de vídeo de múltiples vistas (MVC).

30 **[0012]** HEVC, en general, sigue el marco de las normas de codificación de vídeo anteriores en muchos aspectos. La unidad de predicción en HEVC es diferente de la de ciertas normas de codificación de vídeo anteriores (por ejemplo, macrobloque). De hecho, el concepto de macrobloque no existe en HEVC como se entiende en ciertas normas anteriores de codificación de vídeo. Un macrobloque se reemplaza por una estructura jerárquica basada en un esquema de árbol cuaternario, lo cual puede proporcionar una gran flexibilidad, entre otros beneficios posibles. Por ejemplo, dentro del esquema HEVC, se definen tres tipos de bloques, unidad de codificación (CU), unidad de predicción (PU) y unidad de transformación (TU). La CU puede referirse a la unidad básica de división de la región. La CU puede considerarse análoga al concepto de macrobloque, pero HEVC no limita el tamaño máximo de las CU y puede permitir la división recursiva en cuatro CU de igual tamaño para mejorar la adaptabilidad del contenido. La PU puede considerarse la unidad básica de inter/intrapredicción, y una única PU puede contener múltiples particiones de forma arbitraria para codificar de manera efectiva patrones de imagen irregulares. La TU se puede considerar como la unidad básica de transformación. La TU se puede definir independientemente de la PU; sin embargo, el tamaño de una TU puede estar limitado al tamaño de la CU a la que pertenece la TU. Esta separación de la estructura de bloques en tres conceptos diferentes puede permitir optimizar cada unidad de acuerdo con su función respectiva de la unidad, lo que puede resultar en una mejora de la eficacia de codificación.

45 **[0013]** Para fines de ilustración solamente, ciertos modos de realización divulgados en el presente documento se describen con ejemplos que incluyen solo dos capas (por ejemplo, una capa inferior tal como la capa base y una capa superior tal como la capa de mejora). Debe entenderse que dichos ejemplos pueden ser aplicables a configuraciones que incluyen múltiples capas base y/o de mejora. Además, para facilitar la explicación, la siguiente divulgación incluye los términos "tramas" o "bloques" con referencia a ciertos modos de realización. Sin embargo, estos términos no pretenden ser limitativos. Por ejemplo, las técnicas descritas a continuación se pueden usar con cualquier unidad de vídeo adecuada, como bloques (por ejemplo, CU, PU, TU, macrobloques, etc.), fragmentos, tramas, etc.

Normas de codificación de vídeo

55 **[0014]** Una imagen digital, como una imagen de vídeo, una imagen de TV, una imagen fija o una imagen generada por una grabadora de vídeo o un ordenador puede consistir en píxeles o muestras dispuestas en líneas horizontales y verticales. La cantidad de píxeles en una sola imagen suele ser de decenas de miles. Cada píxel contiene típicamente información de luminancia y crominancia. Sin compresión, la gran cantidad de información que se transmitirá desde un codificador de imágenes a un decodificador de imágenes haría imposible la transmisión de imágenes en tiempo real. Para reducir la cantidad de información a transmitir, se han desarrollado una serie de procedimientos de compresión diferentes, tales como las normas JPEG, MPEG y H.263.

65 **[0015]** Entre las normas de codificación de vídeo se incluyen ITU-T H.261, MPEG-1 Visual de la ISO/IEC, ITU-T H.262 o MPEG-2 Visual de la ISO/IEC, ITU-T H.263, MPEG-4 Visual de la ISO/IEC e ITU-T H.264 (también conocida como AVC del MPEG-4 de la ISO/IEC), incluyendo sus ampliaciones de codificación de vídeo escalable (SVC) y de codificación de vídeo de múltiples vistas (MVC).

5 **[0016]** Además, existe una nueva norma de codificación de vídeo, concretamente la codificación de vídeo de alta eficiencia (HEVC), que está siendo desarrollada por el Equipo de colaboración conjunta en codificación de vídeo (JCT-VC) del Grupo de expertos en codificación de vídeo (VCEG) de ITU-T y el Grupo de expertos en imágenes en movimiento (MPEG) de ISO/IEC. La cita completa del borrador 10 de HEVC es el documento JCTVC-L1003, de Bross y otros, titulado "High Efficiency Video Coding (HEVC) Text Specification Draft 10 [Especificación textual de la codificación de vídeo de alta eficacia (HEVC), Borrador 10]", Equipo de colaboración conjunta en codificación de vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, 12ª conferencia: Ginebra, Suiza, 14 de enero de 2013 al 23 de enero de 2013. La ampliación multivista para HEVC, llamada MV-HEVC, y la ampliación escalable para HEVC, llamada SHVC, también están siendo desarrolladas por el JCT-3V (Equipo de colaboración conjunta de ITU-T/ISO/IEC para el desarrollo de la ampliación de codificación de vídeo 3D) y JCT-VC, respectivamente.

15 **[0017]** A continuación, en el presente documento se describen de forma más detallada diversos aspectos de los sistemas, aparatos y procedimientos novedosos, con referencia a los dibujos adjuntos. Sin embargo, la presente divulgación se puede realizar de muchas formas diferentes y no se debería interpretar que está limitada a ninguna estructura o función específica presentada a lo largo de esta divulgación. En cambio, estos aspectos se proporcionan de modo que esta divulgación sea exhaustiva y completa, y transmita por completo el alcance de la divulgación a los expertos en la materia. Basándose en las enseñanzas del presente documento, un experto en la materia debería apreciar que el alcance de la divulgación pretende abarcar cualquier aspecto de los sistemas, aparatos y procedimientos novedosos divulgados en el presente documento, ya sean implementados de forma independiente de, o en combinación con, cualquier otro aspecto de la presente divulgación. Por ejemplo, un aparato se puede implementar o un procedimiento se puede llevar a la práctica usando cualquier número 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. Debería entenderse que cualquier aspecto divulgado en el presente documento puede realizarse mediante uno o más elementos de una reivindicación.

20 **[0018]** Aunque en el presente documento se describen aspectos particulares, muchas variantes y permutaciones de estos aspectos quedan dentro del alcance de la divulgación. Aunque se mencionan algunos beneficios y ventajas de los aspectos preferentes, el alcance de la divulgación no se pretende limitar a beneficios, usos u objetivos particulares. En cambio, los aspectos de la divulgación están concebidos para ser ampliamente aplicables a diferentes tecnologías inalámbricas, configuraciones de sistema, redes y protocolos de transmisión, algunos de los cuales se ilustran a modo de ejemplo en las figuras y en la siguiente descripción de los aspectos preferentes. La descripción detallada y los dibujos son meramente ilustrativos de la divulgación, en lugar de limitativos, estando el alcance de la divulgación definido por las reivindicaciones adjuntas y los equivalentes de las mismas.

25 **[0019]** Los dibujos adjuntos ilustran ejemplos. Los elementos indicados mediante números de referencia en los dibujos adjuntos corresponden a elementos indicados mediante números de referencia similares en la siguiente descripción. En esta divulgación, los elementos que tienen nombres que comienzan con palabras ordinales (por ejemplo, "primero", "segundo", "tercero", etc.) no necesariamente implican que los elementos tienen un orden particular. Más bien, dichas palabras ordinales se usan simplemente para referirse a diferentes elementos de un mismo tipo o un tipo similar.

45 **Sistema de codificación de vídeo**

50 **[0020]** La **figura 1A** es un diagrama de bloques que ilustra un sistema de codificación de vídeo 10 de ejemplo que puede utilizar técnicas de acuerdo con aspectos descritos en esta divulgación. Tal como se usa en el presente documento, el término "codificador de vídeo" se refiere, en general, tanto a codificadores de vídeo como a descodificadores de vídeo. En esta divulgación, los términos "codificación de vídeo" o "codificación" pueden referirse genéricamente a la codificación de vídeo y a la descodificación de vídeo. Además de los codificadores de vídeo y los descodificadores de vídeo, los aspectos descritos en la presente solicitud pueden extenderse a otros dispositivos relacionados, tales como los transcodificadores (por ejemplo, dispositivos que pueden descodificar un flujo de bits y recodificar otro flujo de bits) y cajas intermedias (por ejemplo, dispositivos que pueden modificar, transformar y/o manipular de otro modo un flujo de bits).

55 **[0021]** Como se muestra en la **figura 1A**, el sistema de codificación de vídeo 10 incluye un módulo de origen 12 que genera datos de vídeo codificados para ser descodificados posteriormente por un módulo de destino 14. En el ejemplo de la **figura 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 **figura 1B**.

60 **[0022]** Con referencia una vez más a la **figura 1A**, el módulo de origen 12 y el módulo de destino 14 pueden comprender cualquiera entre una amplia gama de dispositivos, incluidos los ordenadores de escritorio, los

ordenadores plegables (por ejemplo, los ordenadores portátiles), los ordenadores de tableta, los descodificadores, los equipos telefónicos tales como los llamados "teléfonos inteligentes", los llamados paneles "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivos de transmisión de flujo de vídeo o similares. En algunos casos, el módulo de origen 12 y el módulo de destino 14 pueden estar equipados para la comunicación inalámbrica.

[0023] El módulo de destino 14 puede recibir los datos de vídeo codificados que se van a descodificar a través de un enlace 16. El enlace 16 puede comprender cualquier tipo de medio o dispositivo capaz de mover los datos de vídeo codificados desde el módulo de origen 12 al módulo de destino 14. En el ejemplo de la **figura 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 pueden modularse de acuerdo con una norma 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 cableada, 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.

[0024] De forma alternativa, los datos codificados pueden emitirse desde la 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 mediante una interfaz de entrada 28. El dispositivo de almacenamiento 31 puede incluir cualquiera de una diversidad de medios de almacenamiento de datos, de acceso distribuido o local, tales como una unidad de disco duro, memoria flash, memoria volátil o no volátil u otros medios 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 mediante el módulo de origen 12. El módulo de destino 14 puede acceder a datos de vídeo almacenados del dispositivo de almacenamiento 31, mediante transmisión continua o descarga. El servidor de archivos puede ser cualquier tipo de servidor capaz de almacenar datos de vídeo codificados y transmitir esos datos de vídeo codificados al módulo de destino 14. Los servidores de archivos de ejemplo incluyen un servidor de la Red (por ejemplo, para una sede de la Red), un servidor del 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 mediante cualquier conexión de datos estándar, incluyendo una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión de 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.

[0025] Las técnicas de esta divulgación no están limitadas a aplicaciones o a configuraciones inalámbricas. Las técnicas pueden aplicarse a la codificación de vídeo, para admitir cualquiera de una diversidad de aplicaciones multimedia, tales como radiodifusiones de televisión por el aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de vídeo en continuo, por ejemplo, a través de Internet (por ejemplo, transmisión adaptable dinámica 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 puede configurarse 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.

[0026] En el ejemplo de la **figura 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/desmodulador (módem) y/o un transmisor. En el módulo de origen 12, la fuente de vídeo 18 puede incluir un origen tal como un dispositivo de captura de vídeo, por ejemplo, una videocámara, un archivo de vídeo que contiene vídeo previamente capturado, una interfaz de alimentación de vídeo para recibir vídeo desde un proveedor de contenido de vídeo y/o un sistema de gráficos por ordenador para generar datos de gráficos por ordenador como el vídeo de origen, o una combinación de dichos orígenes. Como un ejemplo, si la fuente de vídeo 18 es una videocámara, el módulo de origen 12 y el módulo de destino 14 pueden formar los denominados "teléfonos de cámara" o "videoteléfonos", tal como se ilustra en el ejemplo de la **figura 1B**. Sin embargo, las técnicas descritas en esta divulgación pueden ser aplicables a la codificación de vídeo en general, y se pueden aplicar a aplicaciones inalámbricas y/o cableadas.

[0027] El vídeo capturado, precapturado o generado por ordenador puede ser codificado por el codificador de vídeo 20. Los datos de vídeo codificados pueden ser transmitidos al módulo de destino 14 a través de la interfaz de salida 22 del módulo de origen 12. Los datos de vídeo codificados pueden almacenarse también (o de forma alternativa) en el dispositivo de almacenamiento 31 para un acceso posterior mediante el módulo de destino 14 u

otros dispositivos, para su descodificación y/o reproducción. El codificador de vídeo 20 ilustrado en la **figura 1A** y la **figura 1B** puede comprender el codificador de vídeo 20 ilustrado en la **figura 2A**, el codificador de vídeo 23 mostrado en la **figura 2B**, o cualquier otro codificador de vídeo descrito en el presente documento.

5 **[0028]** En el ejemplo de la **figura 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 diversidad de elementos sintácticos generados mediante el codificador de vídeo 20, para su uso mediante un descodificador de vídeo, tal como el descodificador de vídeo 30, en la descodificación de los datos de vídeo. Dichos elementos sintácticos se pueden incluir con los datos de vídeo codificados transmitidos en un medio de comunicación, almacenados en un medio de almacenamiento o almacenados en un servidor de archivos. El descodificador de vídeo 30 ilustrado en la **figura 1A** y la **figura 1B** puede comprender el descodificador de vídeo 30 mostrado en la **figura 3A**, el descodificador de vídeo 33 mostrado en la **figura 3B**, o cualquier otro descodificador de vídeo descrito en el presente documento.

20 **[0029]** 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 entre varios dispositivos de visualización, tales como una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodos orgánicos emisores de luz (OLED) u otro tipo de dispositivo de visualización.

25 **[0030]** En aspectos relacionados, la **figura 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 **figura 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 **figura 1B**; sin embargo, en otras implementaciones, la unidad de procesamiento de vídeo 21 puede implementarse como una porción del codificador de vídeo 20 y/o el módulo procesador/controlador 13. El sistema 10' también puede incluir un rastreador 29 opcional, que puede rastrear un objeto de interés en una secuencia de vídeo. El objeto o interés a rastrear puede ser segmentado por una técnica descrita en relación con uno o más aspectos de la presente divulgación. En aspectos relacionados, el rastreo puede ser realizado por el dispositivo de visualización 32, solo o junto con el rastreador 29. El sistema 10' de la **figura 1B**, y sus componentes, son similares de otro modo al sistema 10 de la **figura 1A** y sus componentes.

40 **[0031]** El codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con una norma de compresión de vídeo, tal como la norma de codificación de vídeo de alta eficiencia (HEVC), actualmente en desarrollo, y pueden ajustarse a un modelo de prueba de HEVC (HM). De forma alternativa, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con otras normas de propiedad o industriales, tales como la norma ITU-T H.264, denominada de forma alternativa MPEG-4, parte 10, codificación avanzada de vídeo (AVC), o ampliaciones de dichas normas. Sin embargo, las técnicas de esta divulgación no están limitadas a ninguna norma de codificación particular. Otros ejemplos de normas de compresión de vídeo incluyen MPEG-2 e ITU-T H.263.

50 **[0032]** Aunque no se muestra en los ejemplos de la **figura 1A** y la **figura 1B**, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden estar integrados, cada uno, con un codificador y descodificador de audio, y pueden incluir unidades MUX-DEMUX apropiadas, u otro hardware y software, para gestionar la codificación tanto de audio como de vídeo en un flujo de datos común o flujos de datos por separado. Si procede, en algunos ejemplos, las unidades MUX-DEMUX pueden ajustarse al protocolo de multiplexador ITU H.223 o a otros protocolos, tales como el protocolo de datagramas de usuario (UDP).

55 **[0033]** Cada uno del codificador de vídeo 20 y el descodificador de vídeo 30 pueden implementarse 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 esta divulgación. Cada uno del codificador de vídeo 20 y el descodificador de vídeo 30 pueden estar incluidos en uno o más codificadores o descodificadores, cualquiera de los cuales puede estar integrado como parte de un codificador/descodificador (CÓDEC) combinado en un respectivo dispositivo.

Proceso de codificación de vídeo

[0034] Como se ha mencionado antes brevemente, el codificador de vídeo 20 codifica datos de vídeo. Los datos de vídeo pueden comprender una o más imágenes. Cada una de las imágenes es una imagen fija que forma parte de un vídeo. En algunos casos, una imagen puede denominarse una "trama" de vídeo. Cuando el codificador de vídeo 20 codifica los datos de vídeo, el codificador de vídeo 20 puede generar un flujo de bits. El flujo de bits puede incluir una secuencia de bits que forman una representación codificada de los datos de vídeo. El flujo de bits puede incluir imágenes codificadas y datos asociados. Una imagen codificada es una representación codificada de una imagen.

[0035] Para generar el flujo de bits, el codificador de vídeo 20 puede llevar a cabo operaciones de codificación en cada imagen en los datos de vídeo. Cuando el codificador de vídeo 20 realiza operaciones de codificación en las imágenes, el codificador de vídeo 20 puede generar una serie de imágenes codificadas y datos asociados. Los datos asociados pueden incluir conjuntos de parámetros de vídeo (VPS), conjuntos de parámetros de secuencia, conjuntos de parámetros de imagen, conjuntos de parámetros de adaptación y otras estructuras sintácticas. 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.

[0036] Para generar una imagen codificada, el codificador de vídeo 20 puede dividir una imagen en bloques de vídeo de igual tamaño. Un bloque de vídeo puede ser una matriz bidimensional de muestras. Cada uno de los bloques de vídeo está asociado a un bloque de árbol. En algunos casos, un bloque de árbol también puede denominarse unidad de codificación más grande (LCU). Los bloques de árbol de HEVC pueden ser aproximadamente análogos a los macrobloques de normas anteriores, tales como la H.264/AVC. Sin embargo, un bloque de árbol no está limitado necesariamente a un tamaño particular y puede incluir una o más unidades de codificación (CU). El codificador de vídeo 20 puede usar 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".

[0037] 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.

[0038] Como parte de realizar 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 pueden denominarse "fragmento codificado".

[0039] Para generar un fragmento codificado, el codificador de vídeo 20 puede llevar a cabo operaciones de codificación en cada bloque de árbol en un fragmento. Cuando el codificador de vídeo 20 realiza una operación de codificación en un bloque de árbol, el codificador de vídeo 20 puede generar un bloque de árbol codificado. El bloque de árbol codificado puede comprender datos que representan una versión codificada del bloque de árbol.

[0040] 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 en el fragmento de acuerdo con un orden de escaneo de trama. Por ejemplo, el codificador de vídeo 20 puede codificar los bloques de árbol del fragmento en un orden que avanza de izquierda a derecha en la fila más alta de los bloques de árbol en el fragmento, luego de izquierda a derecha en la siguiente fila inferior de los bloques de árbol, y así sucesivamente hasta que el codificador de vídeo 20 ha codificado cada uno de los bloques de árbol del fragmento.

[0041] Como resultado de codificar los bloques de árbol de acuerdo con el orden de escaneo de trama, los bloques de árbol de arriba y a la izquierda de un bloque de árbol dado pueden haber sido codificados, pero los bloques de árbol de debajo y a la derecha del bloque de árbol dado aún no han sido codificados. En consecuencia, el codificador de vídeo 20 puede tener acceso a la información generada por la codificación de los bloques de árbol de arriba y a la izquierda del bloque de árbol dado cuando codifica el bloque de árbol dado. Sin embargo, el codificador de vídeo 20 puede no tener acceso a la información generada por la codificación de los bloques de árbol de debajo y a la derecha del bloque de árbol dado cuando codifica el bloque de árbol dado.

[0042] Para generar un bloque de árbol codificado, el codificador de vídeo 20 puede llevar a cabo de forma recursiva la división en árbol cuaternario en el bloque de vídeo del bloque de árbol para dividir el bloque de vídeo en bloques de vídeo progresivamente más pequeños. Cada uno de los bloques de vídeo más pequeños puede estar asociado a una CU diferente. Por ejemplo, el codificador de vídeo 20 puede dividir el bloque de vídeo de un bloque de árbol en cuatro subbloques de igual tamaño, dividir uno o más de los subbloques en cuatro

subsubbloques de igual tamaño, etc. Una CU dividida puede ser una CU cuyo bloque de vídeo está dividido en bloques de vídeo asociados a otras CU. Una CU no dividida puede ser una CU cuyo bloque de vídeo no esté dividido en bloques de vídeo asociados a otras CU.

5 **[0043]** Uno o más elementos sintácticos en el flujo de bits pueden indicar un número máximo de veces que el codificador de vídeo 20 puede dividir el bloque de vídeo de un bloque de árbol. Un bloque de vídeo de una CU puede tener forma cuadrada. El tamaño del bloque de vídeo de una CU (es decir, el tamaño de la CU) puede variar desde 8x8 píxeles hasta el tamaño de un bloque de vídeo de un bloque de árbol (es decir, el tamaño del bloque de árbol) con un máximo de 64x64 píxeles o mayor.

10 **[0044]** El codificador de vídeo 20 puede realizar operaciones de codificación en (es decir, codificar) cada CU de un bloque de árbol de acuerdo con un orden de escaneo 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 luego una CU inferior derecha, en ese orden. Cuando el codificador de vídeo 20 realiza una operación de codificación en una CU dividida, el codificador de vídeo 20 puede codificar las CU asociadas a subbloques del bloque de vídeo de la CU dividida de acuerdo con el orden de escaneo 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 después una CU asociada a un subbloque inferior derecho, en ese orden.

20 **[0045]** Como resultado de la codificación de las CU de un bloque de árbol de acuerdo con un orden de escaneo en z, las CU de arriba, de arriba y a la izquierda, de arriba y a la derecha, a la izquierda y de abajo y a la izquierda de una CU dada pueden haber sido codificadas. Las CU de abajo y a la derecha de la CU determinada todavía no han sido codificadas. En consecuencia, el codificador de vídeo 20 puede tener acceso a información generada por la codificación de algunas CU próximas de la CU dada cuando codifica la CU dada. Sin embargo, el codificador de vídeo 20 puede no tener acceso a información generada por la codificación de otras CU próximas de la CU dada cuando codifica la CU dada.

30 **[0046]** Cuando el codificador de vídeo 20 codifica una CU no dividida, el codificador de vídeo 20 puede generar una o más unidades de predicción (PU) para la CU. Cada una de las PU de la CU puede estar asociada a un bloque de vídeo diferente dentro del bloque de vídeo de la CU. El codificador de vídeo 20 puede generar un bloque de vídeo predicho para cada PU de la CU. El bloque de vídeo predicho de una PU puede ser un bloque de muestras. El codificador de vídeo 20 puede usar intrapredicción o interpredicción para generar el bloque de vídeo predicho para una PU.

35 **[0047]** Cuando el codificador de vídeo 20 utiliza la intrapredicción para generar el bloque de vídeo predicho de una PU, el codificador de vídeo 20 puede generar el bloque de vídeo predicho de la PU basándose en muestras descodificadas de la imagen asociada con la PU. Si el codificador de vídeo 20 utiliza la intrapredicción para generar bloques de vídeo predichos de las PU de una CU, la CU es una CU intrapredicha. Si el codificador de vídeo 20 utiliza la interpredicción para generar el bloque de vídeo predicho de la PU, el codificador de vídeo 20 puede generar el bloque de vídeo predicho de la PU basándose en muestras descodificadas de una o más imágenes distintas a la imagen asociada con la PU. Si el codificador de vídeo 20 utiliza la interpredicción para generar bloques de vídeo predichos de las PU de una CU, la CU es una CU interpredicha.

45 **[0048]** Además, cuando el codificador de vídeo 20 utiliza interpredicción para generar un bloque de vídeo predicho para una PU, el codificador de vídeo 20 puede generar información de movimiento para la PU. La información de movimiento para una PU puede indicar uno o más bloques de referencia de la PU. Cada bloque de referencia de la PU puede ser un bloque de vídeo dentro de una imagen de referencia. La imagen de referencia puede ser una imagen distinta de la imagen asociada a la PU. En algunos casos, un bloque de referencia de una PU también puede denominarse "muestra de referencia" de la PU. El codificador de vídeo 20 puede generar el bloque de vídeo predicho para la PU basándose en los bloques de referencia de la PU.

50 **[0049]** Después de que el codificador de vídeo 20 genere bloques de vídeo predichos para una o más PU de una CU, el codificador de vídeo 20 puede generar datos residuales para la CU basándose en los bloques de vídeo predichos para las PU de la CU. Los datos residuales para la CU pueden indicar diferencias entre muestras en los bloques de vídeo predichos para las PU de la CU y el bloque de vídeo original de la CU.

60 **[0050]** Además, como parte de realizar 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 transformación (TU) de la CU. Cada TU de una CU puede estar asociada a un bloque de vídeo residual diferente.

65 **[0051]** El codificador de vídeo 20 puede aplicar una o más transformaciones a los bloques de vídeo residuales asociados con las TU para generar bloques de coeficientes de transformación (por ejemplo, bloques de coeficientes de transformación) asociados con las TU. Conceptualmente, un bloque de coeficientes de transformación puede ser una matriz bidimensional (2D) de coeficientes de transformación.

[0052] Después de generar un bloque de coeficientes de transformación, el codificador de vídeo 20 puede realizar un proceso de cuantificación en el bloque de coeficientes de transformación. La cuantificación se refiere, en general, a un proceso en el que los coeficientes de transformación se cuantifican para reducir, posiblemente, la cantidad de datos usados para representar los coeficientes de transformación, proporcionando una compresión adicional. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos, o a la totalidad, de los coeficientes de transformación. Por ejemplo, un coeficiente de transformación de n bits puede redondearse a la baja hasta un coeficiente de transformación de m bits durante la cuantificación, donde n es mayor que m .

[0053] El codificador de vídeo 20 puede asociar cada CU a un valor de parámetro de cuantificación (QP). El valor QP asociado a una CU puede determinar cómo el codificador de vídeo 20 cuantifica los bloques de coeficientes de transformación asociados a la CU. El codificador de vídeo 20 puede ajustar el grado de cuantificación aplicado a los bloques de coeficientes de transformación asociados a una CU ajustando el valor QP asociado a la CU.

[0054] Después de que el codificador de vídeo 20 cuantifique un bloque de coeficientes de transformación, el codificador de vídeo 20 puede generar conjuntos de elementos sintácticos que representan los coeficientes de transformación en el bloque de coeficientes de transformación cuantificado. El codificador de vídeo 20 puede aplicar operaciones de codificación por entropía, tales como operaciones de codificación aritmética binaria adaptable al contexto (CABAC), a algunos de estos elementos sintácticos. También podrían usarse otras técnicas de codificación por entropía, tales como la codificación de longitud variable adaptable al contexto (CAVLC), la codificación por entropía de división de intervalo de probabilidad (PIPE) u otra codificación aritmética binaria.

[0055] El flujo de bits generado por el codificador de vídeo 20 puede incluir una serie de unidades de capa de abstracción de red (NAL). Cada una de las unidades NAL puede ser una estructura sintáctica que contiene una indicación de un tipo de datos en la unidad NAL y octetos que contienen los datos. Por ejemplo, una unidad NAL puede contener datos que representan un conjunto de parámetros de vídeo, un conjunto de parámetros de secuencia, un conjunto de parámetros de imagen, un fragmento codificado, información de mejora complementaria (SEI), un delimitador de unidad de acceso, datos de relleno u otro tipo de datos. Los datos de una unidad NAL pueden incluir varias estructuras sintácticas.

[0056] 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 sintácticos del flujo de bits. El descodificador de vídeo 30 puede reconstruir las imágenes de los datos de vídeo basándose en los elementos sintácticos extraídos del flujo de bits. El proceso para reconstruir los datos de vídeo basándose en los elementos sintácticos puede ser, en general, recíproco al proceso realizado por el codificador de vídeo 20 para generar los elementos sintácticos.

[0057] Después de que el descodificador de vídeo 30 extraiga los elementos sintácticos asociados con una CU, el descodificador de vídeo 30 puede generar bloques de vídeo predichos para las PU de la CU basándose en los elementos sintácticos. Además, el descodificador de vídeo 30 puede invertir los bloques de coeficientes de transformación cuantificados asociados a las TU de la CU. El descodificador de vídeo 30 puede realizar transformaciones inversas en los bloques de coeficientes de transformación 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 basándose en los bloques de vídeo predichos y los bloques de vídeo residuales. De esta manera, el descodificador de vídeo 30 puede reconstruir los bloques de vídeo de las CU basándose en los elementos sintácticos del flujo de bits.

Codificador de vídeo

[0058] La **figura 2A** es un diagrama de bloques que ilustra un ejemplo de un codificador de vídeo que puede implementar técnicas según aspectos descritos en esta divulgación. El codificador de vídeo 20 puede configurarse para procesar una sola capa de una trama de vídeo, tal como para HEVC. Además, el codificador de vídeo 20 puede configurarse para realizar cualquiera o todas las técnicas de esta divulgación. Como ejemplo, la unidad de procesamiento de predicción 100 puede configurarse para realizar cualquiera o todas las técnicas descritas en esta divulgación. En otro modo de realización, el codificador de vídeo 20 incluye una unidad 128 de predicción entre capas opcional que está configurada para realizar cualquiera o todas las técnicas descritas en esta divulgación. En otros modos de realización, la predicción entre capas se puede realizar mediante la unidad de procesamiento de predicción 100 (por ejemplo, la unidad de interpretación 121 y/o la unidad de intrapredicción 126), en cuyo caso la unidad 128 de predicción entre capas puede omitirse. Sin embargo, los aspectos de esta divulgación no se limitan a lo anterior. En algunos ejemplos, las técnicas descritas en esta divulgación pueden compartirse entre los diversos componentes del codificador de vídeo 20. En algunos ejemplos, adicionalmente o

de forma alternativa, un procesador (no mostrado) puede configurarse para realizar cualquiera o todas las técnicas descritas en esta divulgación.

5 [0059] Con fines de explicación, esta divulgación describe el codificador de vídeo 20 en el contexto de la codificación HEVC. Sin embargo, las técnicas de esta divulgación pueden ser aplicables a otras normas o procedimientos de codificación. El ejemplo representado en la **figura 2A** es para un códec de una sola capa. Sin embargo, como se describirá adicionalmente con respecto a la **figura 2B**, algunos o todos los codificadores de vídeo 20 pueden duplicarse para procesar un códec multicapa.

10 [0060] El codificador de vídeo 20 puede realizar la intracodificación y la intercodificación de bloques de vídeo dentro de fragmentos de vídeo. La intracodificación se basa en la predicción espacial para reducir o eliminar la redundancia espacial en el vídeo dentro de una trama o imagen de vídeo determinada. La intercodificación se basa en la predicción temporal para reducir o eliminar la redundancia temporal en el vídeo dentro de tramas o imágenes adyacentes de una secuencia de vídeo. El intramodo (modo I) puede referirse a cualquiera de varios modos de codificación de base espacial. Los intermodos, tales como la predicción unidireccional (modo P) o la predicción bidireccional (modo B), pueden referirse a cualquiera de varios modos de codificación de base temporal.

15 [0061] En el ejemplo de la **figura 2A**, el codificador de vídeo 20 incluye una pluralidad de componentes funcionales. Los componentes funcionales del codificador de vídeo 20 incluyen una unidad de procesamiento de predicción 100, una unidad de generación residual 102, una unidad de procesamiento de transformación 104, una unidad de cuantificación 106, una unidad de cuantificación inversa 108, una unidad de transformación inversa 110, una unidad de reconstrucción 112, una unidad de filtro 113, una memoria intermedia de imágenes descodificadas 114 y una unidad de codificación por entropía 116. La unidad de procesamiento de predicción 100 incluye una unidad de interpretación 121, una unidad de estimación de movimiento 122, una unidad de compensación de movimiento 124, una unidad de intrapredicción 126 y una unidad de predicción entre capas 128. En otros ejemplos, el codificador de vídeo 20 puede incluir más, menos o diferentes componentes funcionales. Además, la unidad de estimación de movimiento 122 y la unidad de compensación de movimiento 124 pueden estar altamente integradas, pero están representadas en el ejemplo de la **figura 2A** de forma separada con fines explicativos.

20 [0062] 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 **figura 1A** o **1B**) u otra fuente. Los datos de vídeo pueden representar una serie de imágenes. Para codificar los datos de vídeo, el codificador de vídeo 20 puede realizar una operación de codificación en cada una de las imágenes. Como parte de realizar la operación de codificación en una imagen, el codificador de vídeo 20 puede realizar operaciones de codificación en cada fragmento de la imagen. Como parte de realizar una operación de codificación en un fragmento, el codificador de vídeo 20 puede realizar operaciones de codificación en bloques de árbol en el fragmento.

25 [0063] Como parte de realizar una operación de codificación en un bloque de árbol, la unidad de procesamiento de predicción 100 puede realizar la división en árbol cuaternario en el bloque de vídeo del bloque de árbol para dividir el bloque de vídeo en bloques de vídeo progresivamente más pequeños. Cada uno de los bloques de vídeo más pequeños puede estar asociado a una CU diferente. Por ejemplo, la unidad de procesamiento de predicción 100 puede dividir un bloque de vídeo de un bloque de árbol en cuatro subbloques de igual tamaño, dividir uno o más de los subbloques en cuatro subsubbloques de igual tamaño, etc.

30 [0064] Los tamaños de los bloques de vídeo asociados a las CU pueden variar desde muestras de 8x8 hasta el tamaño del bloque de árbol, con un máximo de muestras de 64x64 o mayores. En esta divulgación, "NxN" y "N por N" pueden usarse indistintamente para hacer referencia a las dimensiones de muestras de un bloque de vídeo en lo que respecta a dimensiones verticales y horizontales, por ejemplo, muestras de 16x16 o muestras de 16 por 16. En general, un bloque de vídeo de 16x16 tiene dieciséis muestras en una dirección vertical ($y = 16$) y dieciséis muestras en una dirección horizontal ($x = 16$). Asimismo, un bloque NxN presenta, en general, N muestras en una dirección vertical y N muestras en una dirección horizontal, donde N representa un valor entero no negativo.

35 [0065] Además, como parte de realizar la operación de codificación en un bloque de árbol, la unidad de procesamiento de predicción 100 puede generar una estructura de datos jerárquica en árbol cuaternario para el bloque de árbol. Por ejemplo, un bloque de árbol puede corresponder a un nodo raíz de la estructura de datos en árbol cuaternario. Si la unidad de procesamiento de predicción 100 divide el bloque de vídeo del bloque de árbol en cuatro subbloques, el nodo raíz tiene cuatro nodos secundarios en la estructura de datos en árbol cuaternario. Cada uno de los nodos secundarios corresponde a una CU asociada con uno de los subbloques. Si la unidad de procesamiento de predicción 100 divide uno de los subbloques en cuatro subsubbloques, el nodo correspondiente a la CU asociada con el subbloque puede tener cuatro nodos secundarios, cada uno de los cuales corresponde a una CU asociada con uno de los subsubbloques.

40 [0066] Cada nodo de la estructura de datos en árbol cuaternario puede contener datos sintácticos (por ejemplo, elementos sintácticos) para el bloque de árbol o CU correspondiente. Por ejemplo, un nodo en el árbol cuaternario puede incluir un indicador de división, que indica si el bloque de vídeo de la CU correspondiente al nodo está

dividido (es decir, partido) en cuatro subbloques. Los elementos sintácticos para una CU pueden definirse de manera recursiva y pueden depender de si el bloque de vídeo de la CU está dividido en subbloques. Una CU cuyo bloque de vídeo no está dividido puede corresponder a un nodo hoja en la estructura de datos en árbol cuaternario. Un bloque de árbol codificado puede incluir datos basados en la estructura de datos en árbol cuaternario para un bloque de árbol correspondiente.

[0067] El codificador de vídeo 20 puede realizar operaciones de codificación en cada CU no dividida de un bloque de vídeo 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.

[0068] Como parte de realizar una operación de codificación en una CU, la unidad de procesamiento de predicción 100 puede dividir el bloque de vídeo de la CU entre una o más PU de la CU. El codificador de vídeo 20 y el descodificador de vídeo 30 pueden admitir varios tamaños de PU. Suponiendo que el tamaño de una CU particular sea $2N \times 2N$, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden admitir tamaños de PU de $2N \times 2N$ o $N \times N$ e interpredicción en tamaños de PU simétricas de $2N \times 2N$, $2N \times N$, $N \times 2N$, $N \times N$, $2N \times nU$, $nL \times 2N$, $nR \times 2N$ o similares. El codificador de vídeo 20 y el descodificador de vídeo 30 también pueden admitir 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.

[0069] La unidad de interpredicción 121 puede realizar la interpredicción en cada PU de la CU. La interpredicción puede proporcionar compresión temporal. Para realizar la interpredicción en una PU, la unidad de estimación de movimiento 122 puede generar información de movimiento para la PU. La unidad de compensación de movimiento 124 puede generar un bloque de vídeo predicho para la PU basándose en la información de movimiento y las muestras descodificadas de imágenes distintas a la imagen asociada a la CU (por ejemplo, imágenes de referencia). En esta divulgación, un bloque de vídeo predicho generado por la unidad de compensación de movimiento 124 se puede denominar un bloque de vídeo interpredicho.

[0070] Los fragmentos pueden ser fragmentos I, fragmentos P o fragmentos B. El módulo de estimación de movimiento 122 y la unidad de compensación de movimiento 124 pueden realizar diferentes operaciones para una PU de una CU dependiendo de si la PU está en un fragmento I, un fragmento P o un fragmento B. En un fragmento I, todas las PU se intrapredicen. Por lo tanto, si la PU está en un fragmento I, la unidad de estimación de movimiento 122 y la unidad de compensación de movimiento 124 no realizan la interpredicción en la PU.

[0071] Si la PU está en un fragmento P, la imagen que contiene la PU está asociada a una lista de imágenes de referencia denominada "lista 0". Cada una de las imágenes de referencia de la lista 0 contiene muestras que pueden usarse para la interpredicción de otras imágenes. Cuando la unidad de estimación de movimiento 122 realiza la operación de estimación de movimiento con respecto a una PU en un fragmento P, la unidad de estimación de movimiento 122 puede buscar las imágenes de referencia en la lista 0 para un bloque de referencia para la PU. El bloque de referencia de la PU puede ser un conjunto de muestras, por ejemplo, un bloque de muestras, que se corresponde más estrechamente con las muestras del bloque de vídeo de la PU. La unidad de estimación de movimiento 122 puede usar una variedad de métricas para determinar cuán estrechamente un conjunto de muestras en una imagen de referencia se corresponde con las muestras del bloque de vídeo de una PU. Por ejemplo, la unidad de estimación de movimiento 122 puede determinar cuán estrechamente un conjunto de muestras en una imagen de referencia se corresponde con las muestras en el bloque de vídeo de una PU por la suma de la diferencia absoluta (SAD), la suma de diferencias al cuadrado (SSD) u otras métricas de diferencia.

[0072] Después de identificar un bloque de referencia de una PU en un fragmento P, la unidad de estimación de movimiento 122 puede generar un índice de referencia que indica la imagen de referencia de la lista 0 que contiene el bloque de referencia y un vector de movimiento que indica un desplazamiento espacial entre la PU y el bloque de referencia. En varios ejemplos, la unidad de estimación de movimiento 122 puede generar vectores de movimiento con diferentes grados de precisión. Por ejemplo, la unidad de estimación de movimiento 122 puede generar vectores de movimiento con una precisión de un cuarto de muestra, una precisión de un octavo de muestra o una precisión de otra fracción de muestra. En el caso de la precisión de fracción de muestra, los valores de bloque de referencia pueden interpolarse a partir de valores enteros de posición de muestra en la imagen de referencia. La unidad de estimación de movimiento 122 puede emitir el índice de referencia y el vector de movimiento como la información de movimiento de la PU. La unidad de compensación de movimiento 124 puede generar el bloque de vídeo predicho de la PU basándose en el bloque de referencia identificado por la información de movimiento de la PU.

[0073] 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.

[0074] Además, si la PU está en un fragmento B, la unidad de estimación de movimiento 122 puede llevar a cabo la predicción unidireccional o la predicción bidireccional para la PU. Cuando la unidad de estimación de movimiento

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

[0075] Cuando la unidad de estimación de movimiento 122 realiza la predicción bidireccional para una PU, la unidad de estimación de movimiento 122 puede buscar las imágenes de referencia de la lista 0 para un bloque de referencia para la PU y también puede buscar las imágenes de referencia de la lista 1 para otro bloque de referencia para la PU. La unidad de estimación de movimiento 122 puede entonces generar índices de referencia que indican las imágenes de referencia en la lista 0 y la lista 1 que contienen los bloques de referencia y los vectores de movimiento que indican desplazamientos espaciales entre los bloques de referencia y la PU. La unidad de estimación de movimiento 122 puede emitir los índices de referencia y los vectores de movimiento de la PU como la información de movimiento de la PU. La unidad de compensación de movimiento 124 puede generar el bloque de vídeo predicho de la PU basándose en los bloques de referencia indicados por la información de movimiento de la PU.

[0076] En algunos casos, la unidad de estimación de movimiento 122 no emite un conjunto completo de información de movimiento para una PU a la unidad de codificación por entropía 116. En lugar de eso, la unidad de estimación de movimiento 122 puede señalar la información de movimiento de una PU con referencia a la información de movimiento de otra PU. Por ejemplo, la unidad de estimación de movimiento 122 puede determinar que la información de movimiento de la PU es suficientemente similar a la información de movimiento de una PU próxima. En este ejemplo, la unidad de estimación de movimiento 122 puede indicar, en una estructura sintáctica asociada con la PU, un valor que indica al descodificador de vídeo 30 que la PU tiene la misma información de movimiento que la PU contigua. En otro ejemplo, la unidad de estimación de movimiento 122 puede identificar, en una estructura sintáctica asociada a la PU, una PU próxima y una diferencia de vectores de movimiento (MVD). La diferencia de vectores de movimiento indica una diferencia entre el vector de movimiento de la PU y el vector de movimiento de la PU próxima indicada. El descodificador de vídeo 30 puede utilizar el vector de movimiento de la PU contigua indicada y la diferencia de vectores de movimiento para determinar el vector de movimiento de la PU. Haciendo referencia a la información de movimiento de una primera PU cuando se señala la información de movimiento de una segunda PU, el codificador de vídeo 20 puede ser capaz de señalar la información de movimiento de la segunda PU usando menos bits.

[0077] Como se analizará más adelante a continuación con referencia a las **figuras 8-12**, la unidad de procesamiento de predicción 100 puede configurarse para codificar (por ejemplo, codificar o descodificar) la PU (o cualquier otra capa de referencia y/o bloques de capa de mejora o unidades de vídeo) realizando los procedimientos ilustrados en las **figuras 8-12**. Por ejemplo, la unidad de interpredicción 121 (por ejemplo, a través de la unidad de estimación de movimiento 122 y/o la unidad de compensación de movimiento 124), la unidad de intrapredicción 126, o la unidad de predicción entre capas 128 pueden configurarse para realizar los procedimientos ilustrados en las **figuras 8-12** ya sea juntos o por separado.

[0078] Como parte de realizar una operación de codificación en una CU, la unidad de intrapredicción 126 puede llevar a cabo la intrapredicción en las PU de la CU. La intrapredicción puede proporcionar compresión espacial. Cuando la unidad de intrapredicción 126 realiza la intrapredicción en una PU, la unidad de intrapredicción 126 puede generar datos de predicción para la PU basados en muestras descodificadas de otras PU en la misma imagen. Los datos de predicción para la PU pueden incluir un bloque de vídeo predicho y varios elementos sintácticos. La unidad de intrapredicción 126 puede realizar la intrapredicción en PU en fragmentos I, fragmentos P y fragmentos B.

[0079] Para realizar la intrapredicción en una PU, la unidad de intrapredicción 126 puede usar múltiples modos de intrapredicción para generar múltiples conjuntos de datos de predicción para la PU. Cuando la unidad de intrapredicción 126 utiliza un modo de intrapredicción para generar un conjunto de datos de predicción para la PU, la unidad de intrapredicción 126 puede extender muestras de bloques de vídeo de PU próximas a través del bloque de vídeo de la PU en una dirección y/o gradiente asociados con el modo de intrapredicción. Las PU próximas pueden estar arriba, arriba y a la derecha, arriba y a la izquierda o a la izquierda de la PU, suponiendo un orden de codificación de izquierda a derecha, de arriba a abajo para las PU, CU y bloques de árbol. La unidad de intrapredicción 126 puede usar varios números de modos de intrapredicción, por ejemplo, 33 modos de intrapredicción direccional, dependiendo del tamaño de la PU.

[0080] La unidad de procesamiento de predicción 100 puede seleccionar los datos de predicción para una PU entre los datos de predicción generados por la unidad de compensación de movimiento 124 para la PU o los datos

de predicción generados por la unidad de intrapredicción 126 para la PU. En algunos ejemplos, la unidad de procesamiento de predicción 100 selecciona los datos de predicción para la PU basándose en las métricas de velocidad/distorsión de los conjuntos de datos de predicción.

5 **[0081]** Si la unidad de procesamiento de predicción 100 selecciona datos de predicción generados por la unidad de intrapredicción 126, la unidad de procesamiento de predicción 100 puede señalar el modo de intrapredicción que se utilizó para generar los datos de predicción para las PU, es decir, el modo de intrapredicción seleccionado. La unidad de procesamiento de predicción 100 puede señalar el modo de intrapredicción seleccionado de varias maneras. Por ejemplo, es probable que el modo de intrapredicción seleccionado sea el mismo que el modo de intrapredicción de una PU próxima. En otras palabras, el modo de intrapredicción de la PU próxima puede ser el modo más probable para la PU actual. De este modo, la unidad de procesamiento de predicción 100 puede generar un elemento sintáctico para indicar que el modo de intrapredicción seleccionado es el mismo que el modo de intrapredicción de la PU próxima.

15 **[0082]** Como se analizó anteriormente, el codificador de vídeo 20 puede incluir la unidad de predicción entre capas 128. La unidad de predicción entre capas 128 está configurada para predecir un bloque actual (por ejemplo, un bloque actual en la EL) usando una o más capas diferentes que están disponibles en SVC (por ejemplo, una capa base o de referencia). Dicha predicción se puede denominar predicción entre capas. La unidad de predicción entre capas 128 utiliza procedimientos de predicción para reducir la redundancia entre capas, mejorando así la eficacia de codificación y reduciendo los requisitos de recursos informáticos. Algunos ejemplos de predicción entre capas incluyen la intrapredicción entre capas, la predicción de movimiento entre capas y la predicción residual entre capas. La intrapredicción entre capas utiliza la reconstrucción de bloques ubicados conjuntamente en la capa base para predecir el bloque actual en la capa de mejora. La predicción de movimiento entre capas usa la información de movimiento de la capa base para predecir el movimiento en la capa de mejora. La predicción residual entre capas usa el residuo de la capa base para predecir el residuo de la capa de mejora. Cada uno de los esquemas de predicción entre capas se analiza a continuación con mayor detalle.

30 **[0083]** Después de que la unidad de procesamiento de predicción 100 seleccione los datos de predicción para las PU de una CU, la unidad de generación residual 102 puede generar datos residuales para la CU restando (por ejemplo, indicado por el signo menos) los bloques de vídeo predichos de las PU de la CU del bloque de vídeo de la CU. Los datos residuales de una CU pueden incluir bloques de vídeo residuales 2D que corresponden a diferentes componentes de muestra de las muestras en el bloque de vídeo de la CU. Por ejemplo, los datos residuales pueden incluir un bloque de vídeo residual que corresponde a diferencias entre componentes de luminancia de muestras en los bloques de vídeo predichos de las PU de la CU y componentes de luminancia de muestras en el bloque de vídeo original de la CU. Además, los datos residuales de la CU pueden incluir bloques de vídeo residuales que corresponden a las diferencias entre componentes de crominancia de muestras en los bloques de vídeo predichos de las PU de la CU y los componentes de crominancia de las muestras en el bloque de vídeo original de la CU.

40 **[0084]** La unidad de procesamiento de predicción 100 puede realizar la división en árbol cuaternario para dividir los bloques de vídeo residuales de una CU en subbloques. Cada bloque de vídeo residual no dividido puede estar asociado a una TU diferente de la CU. Los tamaños y posiciones de los bloques de vídeo residuales asociados a las TU de una CU pueden basarse, o no, en los tamaños y posiciones de bloques de vídeo asociados a las PU de la CU. Una estructura en árbol cuaternario conocida como "árbol cuaternario residual" (RQT) puede incluir nodos asociados a cada uno de los bloques de vídeo residuales. Las TU de una CU pueden corresponder a nodos hoja del RQT.

50 **[0085]** La unidad de procesamiento de transformación 104 puede generar uno o más bloques de coeficientes de transformación para cada TU de una CU aplicando una o más transformaciones a un bloque de vídeo residual asociado con la TU. Cada uno de los bloques de coeficientes de transformación puede ser una matriz 2D de coeficientes de transformación. La unidad de procesamiento de transformación 104 puede aplicar varias transformaciones al bloque de vídeo residual asociado con una TU. Por ejemplo, la unidad de procesamiento de transformación 104 puede aplicar una transformación de coseno discreta (DCT), una transformación direccional o una transformación conceptualmente similar al bloque de vídeo residual asociado con una TU.

55 **[0086]** Después de que la unidad de procesamiento de transformación 104 genere un bloque de coeficientes de transformación asociado con una TU, la unidad de cuantificación 106 puede cuantificar los coeficientes de transformación en el bloque de coeficientes de transformación. La unidad de cuantificación 106 puede cuantificar un bloque de coeficientes de transformación asociado con una TU de una CU basándose en un valor de QP asociado con la CU.

65 **[0087]** El codificador de vídeo 20 puede asociar un valor de QP a una CU de varias maneras. Por ejemplo, el codificador de vídeo 20 puede realizar un análisis de velocidad-distorsión en un bloque de árbol asociado 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 varias 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 la velocidad de bits y la métrica de distorsión más bajas.

5 **[0088]** La unidad de cuantificación inversa 108 y la unidad de transformación inversa 110 pueden aplicar la cuantificación inversa y las transformaciones inversas al bloque de coeficientes de transformación, respectivamente, para reconstruir un bloque de vídeo residual a partir del bloque de coeficientes de transformación. La unidad de reconstrucción 112 puede añadir el bloque de vídeo residual reconstruido a las muestras correspondientes de uno o más bloques de vídeo predichos generados por la unidad de procesamiento de predicción 100 para producir un bloque de vídeo reconstruido asociado con una TU. Mediante la reconstrucción de bloques de vídeo para cada TU de una CU de esta manera, el codificador de vídeo 20 puede reconstruir el bloque de vídeo de la CU.

15 **[0089]** Después de que la unidad de reconstrucción 112 reconstruya el bloque de vídeo de una CU, la unidad de filtro 113 puede realizar una operación de desbloqueo para reducir los artefactos de bloqueo en el bloque de vídeo asociado con la CU. Después de realizar las una o más operaciones de desbloqueo, la unidad de filtro 113 puede almacenar el bloque de vídeo reconstruido de la CU en la memoria intermedia de imágenes descodificadas 114. La unidad de estimación de movimiento 122 y la unidad de compensación de movimiento 124 pueden utilizar una imagen de referencia que contiene el bloque de vídeo reconstruido para realizar la interpredicción en las PU de las imágenes posteriores. Además, la unidad de intrapredicción 126 puede usar bloques de vídeo reconstruidos en la memoria intermedia de imágenes descodificadas 114 para realizar la intrapredicción en otras PU en la misma imagen que la CU.

25 **[0090]** 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 la unidad de cuantificación 106 y puede recibir elementos sintácticos de la unidad de procesamiento de predicción 100. Cuando la unidad de codificación por entropía 116 recibe los datos, la unidad de codificación por entropía 116 puede realizar una o más operaciones de codificación por entropía para generar datos codificados por entropía. Por ejemplo, el codificador de vídeo 20 puede realizar una operación de codificación de longitud variable adaptable al contexto (CAVLC), una operación CABAC, una operación de codificación de longitud variable a variable (V2V), una operación de codificación aritmética binaria adaptable al contexto basada en sintaxis (SBAC), una operación de codificación por entropía de división de intervalo de probabilidad (PIPE) u otro tipo de operación de codificación por entropía en los datos. La unidad de codificación por entropía 116 puede emitir un flujo de bits que incluye los datos codificados por entropía.

35 **[0091]** Como parte de realizar una operación de codificación por entropía en los datos, la unidad de codificación por entropía 116 puede seleccionar un modelo de contexto. Si la unidad de codificación por entropía 116 está realizando una operación CABAC, el modelo de contexto puede indicar las estimaciones de las probabilidades de que un bin particular tenga unos valores particulares. En el contexto de CABAC, el término "bin" se utiliza para hacer referencia a un bit de una versión binarizada de un elemento sintáctico.

Codificador de vídeo de múltiples capas

45 **[0092]** La **figura 2** es un diagrama de bloques que ilustra un ejemplo de un codificador de vídeo de múltiples capas 23 que puede implementar técnicas de acuerdo con los aspectos descritos en esta divulgación. El codificador de vídeo 23 puede configurarse para procesar tramas de vídeo de múltiples capas, como para codificación SHVC y multivista. Además, el codificador de vídeo 23 puede configurarse para realizar cualquiera o todas las técnicas de esta divulgación.

50 **[0093]** El codificador de vídeo 23 incluye un codificador de vídeo 20A y un codificador de vídeo 20B, cada uno de los cuales puede configurarse como el codificador de vídeo 20 y puede realizar las funciones descritas anteriormente con respecto al codificador de vídeo 20. Además, como se indica por la reutilización de los números de referencia, los codificadores de vídeo 20A y 20B pueden incluir, al menos, algunos de los sistemas y subsistemas como el codificador de vídeo 20. Aunque el codificador de vídeo 23 se ilustra como que incluye dos codificadores de vídeo 20A y 20B, el codificador de vídeo 23 no está limitado como tal y puede incluir cualquier número de capas de codificador de vídeo 20. En algunos modos de realización, el codificador de vídeo 23 puede incluir un codificador de vídeo 20 para cada imagen o trama en una unidad de acceso. Por ejemplo, una unidad de acceso que incluye cinco imágenes puede ser procesada o codificada por un codificador de vídeo que incluye cinco capas de codificador. En algunos modos de realización, el codificador de vídeo 23 puede incluir más capas de codificador que tramas en una unidad de acceso. En algunos de dichos casos, algunas de las capas del codificador de vídeo pueden estar inactivas al procesar algunas unidades de acceso.

65 **[0094]** 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, interpolar una capa base de una trama de vídeo recibida para, por ejemplo, crear una capa de mejora. La unidad de remuestreo 90 puede interpolar la información particular asociada con la capa base recibida de una trama, pero no otra información. Por ejemplo, la

unidad de remuestreo 90 puede interpolar el tamaño espacial o el número de píxeles de la capa base, pero el número de fragmentos o el recuento de orden de imágenes puede permanecer constante. En algunos casos, la unidad de remuestreo 90 puede no procesar el vídeo recibido y/o puede ser opcional. Por ejemplo, en algunos casos, la unidad de procesamiento de predicción 100 puede realizar una interpolación. En algunos modos de realización, la unidad de remuestreo 90 está configurada para 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 escaneo 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 diezmar una capa. Por ejemplo, si durante la transmisión de un vídeo se reduce el ancho de banda, una trama puede ser diezmada en lugar de interpolada.

[0095] La unidad de remuestreo 90 puede configurarse para recibir una imagen o trama (o información de imagen asociada con la imagen) desde la memoria intermedia de imágenes descodificadas 114 del codificador de capa inferior (por ejemplo, el codificador de vídeo 20A) y para interpolar la imagen (o la información de imagen recibida). Esta imagen interpolada puede entonces proporcionarse a la unidad de procesamiento de predicción 100 de un codificador de capa superior (por ejemplo, el codificador de vídeo 20B) configurado para codificar una imagen en la misma unidad de acceso que el codificador de capa inferior. En algunos casos, el codificador de capa superior es una capa eliminada del codificador de capa inferior. En otros casos, puede haber uno o más codificadores de capa superior entre el codificador de vídeo de capa 0 y el codificador de capa 1 de la **figura 2B**.

[0096] En algunos casos, la unidad de remuestreo 90 se puede omitir o anular. En dichos casos, la imagen de la memoria intermedia de imágenes descodificadas 114 del codificador de vídeo 20A se puede proporcionar directamente, o al menos sin proporcionarse a la unidad de remuestreo 90, a la unidad de procesamiento de predicción 100 del codificador de vídeo 20B. Por ejemplo, si los datos de vídeo proporcionados al codificador de vídeo 20B y la imagen de referencia de la memoria intermedia de imágenes descodificadas 114 del codificador de vídeo 20A son del mismo tamaño o resolución, la imagen de referencia puede proporcionarse al codificador de vídeo 20B sin ningún remuestreo.

[0097] En algunos modos de realización, el codificador de vídeo 23 diezma los datos de vídeo a proporcionar al codificador de capa inferior usando la unidad de diezmo 94 antes de proporcionar los datos de vídeo al codificador de vídeo 20A. De forma alternativa, la unidad de diezmo 94 puede ser una unidad de remuestreo 90 capaz de interpolar o diezmar los datos de vídeo. En otros modos de realización más, la unidad de diezmo 94 puede omitirse.

[0098] Como se ilustra en la **figura 2B**, el codificador de vídeo 23 puede incluir, además, un multiplexor 98, o mux. El mux 98 puede emitir 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 emite en un momento determinado. Mientras que en algunos casos los bits de los dos (o más en el caso de más de dos capas de codificador de vídeo) flujos de bits pueden alternarse un bit a la vez, en muchos casos los flujos de bits se combinan de manera diferente. Por ejemplo, el flujo de bits de salida puede crearse alternando el flujo de bits seleccionado un bloque a la vez. En otro ejemplo, el flujo de bits de salida se puede crear emitiendo una proporción de bloques diferente a 1:1 de cada uno de los codificadores de vídeo 20A y 20B. Por ejemplo, pueden emitirse dos bloques desde el codificador de vídeo 20B para cada salida de bloque del codificador de vídeo 20A. En algunos modos de realización, el flujo de salida del mux 98 puede preprogramarse. En otros modos de realización, el mux 98 puede combinar los flujos de bits de los codificadores de vídeo 20A, 20B en base a una señal de control recibida desde un sistema externo al codificador de vídeo 23, tal como desde un procesador en un dispositivo de origen que incluye el módulo de origen 12. La señal de control puede generarse en base a la resolución o velocidad de bits de un vídeo desde la fuente de vídeo 18, en base al ancho de banda del enlace 16, en base a una suscripción asociada a un usuario (por ejemplo, una suscripción de pago frente a una suscripción gratuita) o en base a cualquier otro factor para determinar una salida de resolución deseada del codificador de vídeo 23.

Descodificador de vídeo

[0099] La **figura 3A** es un diagrama de bloques que ilustra un ejemplo de un descodificador de vídeo que puede implementar técnicas según aspectos descritos en esta divulgación. El descodificador de vídeo 30 puede configurarse 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 cualquiera o todas las técnicas de esta divulgación. Como ejemplo, la unidad de compensación de movimiento 162 y/o la unidad de intrapredicción 164 pueden configurarse para realizar cualquiera o todas las técnicas descritas en esta divulgación. En un modo de realización, el descodificador de vídeo 30 puede incluir opcionalmente una unidad de predicción 166 entre capas que está configurada para realizar cualquiera o todas las técnicas descritas en esta divulgación. En otros modos de realización, la predicción entre capas puede realizarse mediante la unidad de procesamiento de predicción 152 (por ejemplo, la unidad de compensación de movimiento 162 y/o la unidad de intrapredicción 164), en cuyo caso la unidad de predicción entre capas 166 puede omitirse. Sin embargo, los aspectos de esta divulgación no se limitan a lo anterior. En algunos ejemplos, las técnicas descritas en esta divulgación pueden compartirse entre los diversos componentes del

descodificador de vídeo 30. En algunos ejemplos, adicionalmente o de forma alternativa, un procesador (no mostrado) puede configurarse para realizar cualquiera o todas las técnicas descritas en esta divulgación.

5 **[0100]** Con fines explicativos, esta divulgación describe un descodificador de vídeo 30 en el contexto de la codificación HEVC. Sin embargo, las técnicas de esta divulgación pueden ser aplicables a otras normas o procedimientos de codificación. El ejemplo representado en la **figura 3A** es para un códec de una sola capa. Sin embargo, como se describirá adicionalmente con respecto a la **figura 3B**, parte o la totalidad del descodificador de vídeo 30 puede duplicarse para procesar un códec multicapa.

10 **[0101]** En el ejemplo de la **figura 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
15 unidad de transformación inversa 156, una unidad de reconstrucción 158, una unidad de filtro 159 y una memoria intermedia de imágenes descodificadas 160. La unidad de procesamiento de predicción 152 incluye una unidad de compensación de movimiento 162, una unidad de intrapredicción 164 y una unidad de predicción entre capas 166. En algunos ejemplos, el 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 **figura 2A**. En otros ejemplos, el descodificador de vídeo 30 puede incluir más, menos o diferentes componentes funcionales.

20 **[0102]** 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 sintácticos. 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 sintácticos del flujo de bits. Como parte de realizar la
25 operación de análisis sintáctico, la unidad de descodificación por entropía 150 puede realizar la descodificación por entropía de elementos sintácticos sometidos a codificación por entropía en el flujo de bits. La unidad de procesamiento de predicción 152, la unidad de cuantificación inversa 154, la unidad de transformación inversa 156, la unidad de reconstrucción 158 y la unidad de filtro 159 pueden realizar una operación de reconstrucción que genera datos de vídeo descodificados basados en los elementos sintácticos extraídos del flujo de bits.

30 **[0103]** Como se analizó anteriormente, el flujo de bits puede comprender una serie de unidades NAL. Las unidades NAL del flujo de bits pueden incluir unidades NAL de conjunto de parámetros de vídeo, unidades NAL de conjunto de parámetros de secuencia, unidades NAL de conjunto de parámetros de imagen, unidades NAL SEI, etc. Como parte de realizar la operación de análisis sintáctico en el flujo de bits, la unidad de descodificación por entropía 150 puede realizar operaciones de análisis sintáctico que extraen y descodifican por entropía conjuntos
35 de parámetros de secuencia a partir de unidades NAL de conjuntos de parámetros de secuencia, conjuntos de parámetros de imagen a partir de unidades NAL de conjuntos de parámetros de imagen, datos SEI a partir de unidades NAL SEI, etc.

40 **[0104]** Además, las unidades NAL del flujo de bits pueden incluir unidades NAL de fragmentos codificados. Como parte de realizar la operación de análisis sintáctico en el flujo de bits, la unidad de descodificación por entropía 150 puede realizar operaciones de análisis sintáctico que extraen y descodifican por entropía fragmentos codificados a partir de las unidades de NAL de fragmentos codificados. Cada uno de los fragmentos codificados puede incluir una cabecera de fragmento y datos de fragmento. La cabecera de fragmento puede contener elementos sintácticos
45 pertenecientes a un fragmento. Los elementos sintácticos de la cabecera de fragmento pueden incluir un elemento sintáctico que identifica un conjunto de parámetros de imagen asociado a una imagen que contiene el fragmento. La unidad de descodificación por entropía 150 puede realizar operaciones de descodificación por entropía, tales como operaciones de descodificación de CABAC, en los elementos sintácticos de la cabecera del fragmento codificado para recuperar la cabecera del fragmento.

50 **[0105]** Como parte de extraer los datos del fragmento de unidades NAL de fragmentos codificados, la unidad de descodificación por entropía 150 puede realizar operaciones de análisis sintáctico que extraen los elementos sintácticos de las CU codificadas en los datos del fragmento. Los elementos sintácticos extraídos pueden incluir elementos sintácticos asociados a bloques de coeficientes de transformación. La unidad de descodificación por entropía 150 puede entonces realizar operaciones de descodificación de CABAC en algunos de los elementos
55 sintácticos.

60 **[0106]** Después de que la unidad de descodificación por entropía 150 realice una operación de análisis sintáctico en una CU no dividida, el descodificador de vídeo 30 puede realizar una operación de reconstrucción en la CU no dividida. Para realizar la operación de reconstrucción en una CU no dividida, el descodificador de vídeo 30 puede realizar una operación de reconstrucción en cada TU de la CU. 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.

65 **[0107]** Como parte de realizar una operación de reconstrucción en una TU, la unidad de cuantificación inversa 154 puede cuantificar de forma inversa, es decir, descuantificar, un bloque de coeficientes de transformación asociado a la TU. La unidad de cuantificación inversa 154 puede cuantificar de forma inversa el bloque de

coeficientes de transformación de una manera similar a los procesos de cuantificación inversa propuestos para HEVC o definidos por la norma de descodificación H.264. La unidad de cuantificación inversa 154 puede utilizar un parámetro de cuantificación QP calculado por el codificador de vídeo 20 para una CU del bloque de coeficientes de transformación para determinar un grado de cuantificación y, del mismo modo, un grado de cuantificación inversa para la unidad de cuantificación inversa 154 a aplicar.

[0108] Después de que la unidad de cuantificación inversa 154 cuantifique de forma inversa un bloque de coeficientes de transformación, la unidad de transformación inversa 156 puede generar un bloque de vídeo residual para la TU asociada con el bloque de coeficientes de transformación. La unidad de transformación inversa 156 puede aplicar una transformación inversa al bloque de coeficientes de transformación para generar el bloque de vídeo residual para la TU. Por ejemplo, la unidad de transformación inversa 156 puede aplicar una DCT inversa, una transformación de número entero inversa, una transformación de Karhunen-Loeve (KLT) inversa, una transformación de rotación inversa, una transformación direccional inversa u otra transformación inversa al bloque de coeficientes de transformación. En algunos ejemplos, la unidad de transformación inversa 156 puede determinar una transformación inversa para aplicar al bloque de coeficientes de transformación basándose en la señalización del codificador de vídeo 20. En dichos ejemplos, la unidad de transformación inversa 156 puede determinar la transformación inversa basándose en una transformación señalizada en el nodo raíz de un árbol cuaternario para un bloque de árbol asociado al bloque de coeficientes de transformación. En otros ejemplos, la unidad de transformación inversa 156 puede inferir la transformación inversa a partir de una o más características de codificación, tales como tamaño de bloque, modo de codificación o similares. En algunos ejemplos, la unidad de transformación inversa 156 puede aplicar una transformación inversa en cascada.

[0109] En algunos ejemplos, la unidad de compensación de movimiento 162 puede refinar el bloque de vídeo predicho de una PU mediante interpolación basándose en filtros de interpolación. Los identificadores para los filtros de interpolación que van a usarse para la compensación de movimiento con una precisión de submuestra pueden incluirse en los elementos sintácticos. La unidad de compensación de movimiento 162 puede usar los mismos filtros de interpolación usados por el codificador de vídeo 20 durante la generación del bloque de vídeo predicho de la PU para calcular valores interpolados para muestras de subenteros de un bloque de referencia. La unidad de compensación de movimiento 162 puede determinar los filtros de interpolación usados por el codificador de vídeo 20 de acuerdo con la información sintáctica recibida y usar los filtros de interpolación para producir el bloque de vídeo predicho.

[0110] Como se analiza adicionalmente más adelante con referencia a las **figuras 8-12**, la unidad de procesamiento de predicción 152 puede codificar (por ejemplo, codificar o descodificar) la PU (o cualquier otra capa de referencia y/o bloques de capa de mejora o unidades de vídeo) realizando los procedimientos ilustrados en las **figuras 8-12**. Por ejemplo, la unidad de compensación de movimiento 162, la unidad de intrapredicción 164 o la unidad de predicción entre capas 166 pueden configurarse para realizar los procedimientos ilustrados en las **figuras 8-12**, ya sea juntos o por separado.

[0111] Si se codifica una PU usando intrapredicción, la unidad de intrapredicción 164 puede realizar la intrapredicción para generar un bloque de vídeo predicho para la PU. Por ejemplo, la unidad de intrapredicción 164 puede determinar un modo de intrapredicción para la PU basándose en los elementos sintácticos del flujo de bits. El flujo de bits puede incluir elementos sintácticos que la unidad de intrapredicción 164 puede usar para determinar el modo de intrapredicción de la PU.

[0112] En algunos casos, los elementos sintácticos pueden indicar que la unidad de intrapredicción 164 tiene que utilizar el modo de intrapredicción de otra PU para determinar el modo de intrapredicción de la PU actual. Por ejemplo, puede ser probable que el modo de intrapredicción de la PU actual sea el mismo que el modo de intrapredicción de una PU próxima. En otras palabras, el modo de intrapredicción de la PU próxima puede ser el modo más probable para la PU actual. Por lo tanto, en este ejemplo, el flujo de bits puede incluir un pequeño elemento sintáctico que indica que el modo de intrapredicción de la PU es el mismo que el modo de intrapredicción de la PU próxima. La unidad de intrapredicción 164 puede entonces utilizar el modo de intrapredicción para generar datos de predicción (por ejemplo, muestras predichas) para la PU basándose en los bloques de vídeo de las PU espacialmente próximas.

[0113] Como se analizó anteriormente, el descodificador de vídeo 30 también puede incluir la unidad de predicción entre capas 166. La unidad de predicción entre capas 166 está configurada para predecir un bloque actual (por ejemplo, un bloque actual en la EL) usando una o más capas diferentes que están disponibles en SVC (por ejemplo, una capa base o de referencia). Dicha predicción se puede denominar predicción entre capas. La unidad de predicción entre capas 166 utiliza procedimientos de predicción para reducir la redundancia entre capas, mejorando así la eficacia de codificación y reduciendo los requisitos de recursos informáticos. Algunos ejemplos de predicción entre capas incluyen la intrapredicción entre capas, la predicción de movimiento entre capas y la predicción residual entre capas. La intrapredicción entre capas utiliza la reconstrucción de bloques ubicados conjuntamente en la capa base para predecir el bloque actual en la capa de mejora. La predicción de movimiento entre capas usa la información de movimiento de la capa base para predecir el movimiento en la capa de mejora.

La predicción residual entre capas usa el residuo de la capa base para predecir el residuo de la capa de mejora. Cada uno de los esquemas de predicción entre capas se analiza a continuación con mayor detalle.

5 [0114] La unidad de reconstrucción 158 puede utilizar los bloques de vídeo residuales asociados a las TU de una CU y los bloques de vídeo predichos de las PU de la CU, es decir, datos de intrapredicción o datos de interpredicción, según corresponda, para reconstruir el bloque de vídeo de la CU. De este modo, el descodificador de vídeo 30 puede generar un bloque de vídeo predicho y un bloque de vídeo residual basándose en los elementos sintácticos en el flujo de bits y puede generar un bloque de vídeo basándose en el bloque de vídeo predicho y el bloque de vídeo residual.

10 [0115] Después de que la unidad de reconstrucción 158 reconstruya el bloque de vídeo de la CU, la unidad de filtro 159 puede realizar una operación de desbloqueo para reducir los artefactos de bloqueo asociados a la CU. Después de que la unidad de filtro 159 realice una operación de desbloqueo para reducir los artefactos de bloqueo asociados con la CU, el descodificador de vídeo 30 puede almacenar el bloque de vídeo de la CU en la memoria intermedia de imágenes descodificadas 160. La memoria intermedia de imágenes descodificadas 160 puede proporcionar imágenes de referencia para una posterior compensación de movimiento, intrapredicción y presentación en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la **figura 1A** o **1B**. Por ejemplo, el descodificador de vídeo 30 puede realizar, basándose en los bloques de vídeo en la memoria intermedia de imágenes descodificadas 160, operaciones de intrapredicción o de interpredicción en las PU de otras CU.

Descodificador de múltiples capas

25 [0116] La **figura 3B** es un diagrama de bloques que ilustra un ejemplo de un descodificador de vídeo 33 de múltiples capas que puede implementar técnicas de acuerdo con aspectos descritos en esta divulgación. El descodificador de vídeo 33 puede configurarse para procesar tramas de vídeo de múltiples capas, como para codificación SHVC y multivista. Además, el descodificador de vídeo 33 se puede configurar para realizar cualquiera o todas las técnicas de esta divulgación.

30 [0117] El descodificador de vídeo 33 incluye un descodificador de vídeo 30A y un descodificador de vídeo 30B, cada uno de los cuales puede configurarse como el descodificador de vídeo 30 y puede realizar las funciones descritas anteriormente con respecto al descodificador de vídeo 30. Además, como se indica por la reutilización de los números de referencia, los descodificadores de vídeo 30A y 30B pueden incluir al menos algunos de los sistemas y subsistemas como el descodificador de vídeo 30. Aunque el descodificador de vídeo 33 se ilustra como que incluye dos descodificadores de vídeo 30A y 30B, el descodificador de vídeo 33 no está limitado como tal y puede incluir cualquier número de capas de descodificador de vídeo 30. En algunos modos de realización, el descodificador de vídeo 33 puede incluir un descodificador de vídeo 30 para cada imagen o trama en una unidad de acceso. Por ejemplo, una unidad de acceso que incluye cinco imágenes puede procesarse o descodificarse mediante un descodificador de vídeo que incluye cinco capas de descodificador. En algunos modos de realización, el descodificador de vídeo 33 puede incluir más capas de descodificador que tramas en una unidad de acceso. En algunos de dichos casos, algunas de las capas del descodificador de vídeo pueden estar inactivas al procesar algunas unidades de acceso.

45 [0118] 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 para ser añadida a la lista de imágenes de referencia para la trama o la unidad de acceso. Esta capa mejorada se puede almacenar en la memoria intermedia de imágenes descodificadas 160. En algunos modos de realización, la unidad de interpolación 92 puede incluir algunos o todos los modos de realización descritos con respecto a la unidad de remuestreo 90 de la **figura 2A**. En algunos modos de realización, la unidad de interpolación 92 está configurada 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 escaneo de trama. En algunos casos, la unidad de interpolación 92 puede ser una unidad de remuestreo configurada para interpolar y/o diezmar una capa de una trama de vídeo recibida.

55 [0119] La unidad de interpolación 92 puede estar configurada para recibir una imagen o trama (o información de imagen asociada a la imagen) de la memoria intermedia de imágenes descodificadas 160 del descodificador de capa inferior (por ejemplo, el descodificador de vídeo 30A) y para interpolar la imagen (o la información de imagen recibida). Esta imagen interpolada puede entonces proporcionarse a la unidad de procesamiento de predicción 152 de un descodificador de capa superior (por ejemplo, el descodificador de vídeo 30B) configurado para descodificar una imagen en la misma unidad de acceso que el descodificador de capa inferior. En algunos casos, el descodificador de capa superior es una capa eliminada del descodificador de capa inferior. En otros casos, puede haber uno o más descodificadores de capa superior entre el descodificador de capa 0 y el descodificador de capa 1 de la **figura 3B**.

65 [0120] En algunos casos, la unidad de interpolación 92 se puede omitir o anular. En dichos casos, la imagen de la memoria intermedia de imágenes descodificadas 160 del descodificador de vídeo 30A se puede proporcionar

directamente, o al menos sin proporcionarse a la unidad de interpolación 92, a la unidad de procesamiento de predicción 152 del descodificador de vídeo 30B. Por ejemplo, si los datos de vídeo proporcionados al descodificador de vídeo 30B y la imagen de referencia de la memoria intermedia de imágenes descodificadas 160 del descodificador de vídeo 30A son del mismo tamaño o resolución, la imagen de referencia puede proporcionarse al descodificador de vídeo 30B sin interpolación. Además, en algunos modos de realización, la unidad de interpolación 92 puede ser una unidad de remuestreo 90 configurada para interpolar o diezmar la imagen de referencia recibida de la memoria intermedia de imágenes descodificadas 160 del descodificador de vídeo 30A.

[0121] Tal como se ilustra en la **figura 3B**, el descodificador de vídeo 33 puede incluir, además, un demultiplexor 99, o demux. El demux 99 puede dividir un flujo de bits de vídeo codificado en múltiples flujos de bits con cada flujo de bits emitido por el demux 99 que se proporciona a un descodificador de vídeo diferente 30A y 30B. Los múltiples flujos de bits pueden crearse al recibir un flujo de bits y cada uno de los descodificadores de vídeo 30A y 30B recibe una porción del flujo de bits en un momento determinado. Mientras que en algunos casos los bits del flujo de bits recibidos en el demux 99 pueden alternarse un bit a la vez entre cada uno de los descodificadores de vídeo (por ejemplo, los descodificadores de vídeo 30A y 30B en el ejemplo de la **figura 3B**), en muchos casos el flujo de bits se divide de manera diferente. Por ejemplo, el flujo de bits se puede dividir alternando qué descodificador de vídeo recibe el flujo de bits un bloque a la vez. En otro ejemplo, el flujo de bits puede dividirse por una proporción de bloques diferente a 1: 1 a cada uno de los descodificadores de vídeo 30A y 30B. Por ejemplo, pueden proporcionarse dos bloques al descodificador de vídeo 30B por cada bloque proporcionado al descodificador de vídeo 30A. En algunos modos de realización, la división del flujo de bits por el demux 99 puede preprogramarse. En otros modos de realización, el demux 99 puede dividir el flujo de bits en base a una señal de control recibida desde un sistema externo al descodificador de vídeo 33, tal como desde un procesador en un dispositivo de destino que incluye el módulo de destino 14. La señal de control puede generarse en base a la resolución o la velocidad de bits de un vídeo desde la interfaz de entrada 28, en base al ancho de banda del enlace 16, en base a una suscripción asociada a un usuario (por ejemplo, una suscripción de pago frente a una suscripción gratuita) o en base a cualquier otro factor para determinar una resolución obtenible por el descodificador de vídeo 33.

Imágenes de punto de acceso aleatorio interno (IRAP)

[0122] Algunos esquemas de codificación de vídeo pueden proporcionar diversos puntos de acceso aleatorios en todo el flujo de bits de manera que el flujo de bits puede descodificarse comenzando desde cualquiera de esos puntos de acceso aleatorios sin necesidad de descodificar ninguna imagen que preceda esos puntos de acceso aleatorios en el flujo de bits. En dichos esquemas de codificación de vídeo, todas las imágenes que siguen un punto de acceso aleatorio en el orden de salida (por ejemplo, incluyendo aquellas imágenes que están en la misma unidad de acceso que la imagen que proporciona el punto de acceso aleatorio) se pueden descodificar correctamente sin usar ninguna imagen que preceda al punto de acceso aleatorio. Por ejemplo, incluso si una parte del flujo de bits se pierde durante la transmisión o durante la descodificación, un descodificador puede reanudar la descodificación del flujo de bits comenzando desde el próximo punto de acceso aleatorio. La compatibilidad con el acceso aleatorio puede facilitar, por ejemplo, servicios dinámicos de transmisión, operaciones de búsqueda, cambio de canal, etc.

[0123] En algunos esquemas de codificación, dichos puntos de acceso aleatorio pueden proporcionarse mediante imágenes que se denominan imágenes de punto de acceso aleatorio interno (IRAP). Por ejemplo, un punto de acceso aleatorio (por ejemplo, proporcionado por una imagen IRAP de capa de mejora) en una capa de mejora ("capaA") que está contenida en una unidad de acceso ("auA") puede proporcionar acceso aleatorio específico de capa tal que para cada capa de referencia ("capaB") de la capaA (por ejemplo, una capa de referencia que es una capa que se usa para predecir la capaA) que tiene un punto de acceso aleatorio contenido en una unidad de acceso ("auB") que está en la capaB y precede a la auA en el orden de descodificación (o un punto de acceso aleatorio contenido en la auA), las imágenes en la capaA que siguen a la auB en el orden de salida (incluidas las imágenes ubicadas en la auB) se pueden descodificar correctamente sin necesidad de descodificar ninguna imagen en la capaA que preceda a la auB.

[0124] Las imágenes IRAP pueden codificarse usando la intrapredicción (por ejemplo, codificarse sin hacer referencia a otras imágenes), y pueden incluir, por ejemplo, imágenes instantáneas de actualización 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 el orden de descodificación no se utilizan para la predicción mediante las imágenes que siguen a la imagen IDR en el orden de descodificación. Cuando hay una imagen CRA en el flujo de bits, las imágenes que siguen a la imagen CRA pueden o no usar imágenes que preceden a la imagen CRA en el orden de descodificación para la predicción. Dichas imágenes que siguen a la imagen CRA en el orden de descodificación, pero usan imágenes que preceden a la imagen CRA en el orden de descodificación, pueden denominarse imágenes principales omitidas de acceso aleatorio (RASL). Otro tipo de imagen que sigue a una imagen IRAP en el orden de descodificación y que precede a la imagen IRAP en el orden de salida es una imagen principal descodificable de acceso aleatorio (RADL), que puede no contener referencias a ninguna imagen que preceda a la imagen IRAP en el orden de descodificación. Las imágenes RASL pueden ser descartadas por el descodificador si las imágenes que preceden a la imagen CRA no están disponibles. Una imagen BLA indica al descodificador que las imágenes que preceden a la imagen BLA

pueden no estar disponibles para el descodificador (por ejemplo, porque dos flujos de bits están unidos entre sí y la imagen BLA es la primera imagen del segundo flujo de bits en el orden de descodificación). Una unidad de acceso (por ejemplo, un grupo de imágenes que consisten en todas las imágenes codificadas asociadas con el mismo tiempo de salida en las múltiples capas) que contiene una imagen de capa base (por ejemplo, una imagen que tiene un valor de identificador (ID) de capa de 0) que es una imagen IRAP se puede denominar una unidad de acceso IRAP.

Alineación de capas cruzadas de imágenes IRAP

[0125] En SVC, es posible que no se requiera que las imágenes IRAP estén alineadas (por ejemplo, contenidas en la misma unidad de acceso) en diferentes capas. Por ejemplo, si las imágenes IRAP debían estar alineadas, cualquier unidad de acceso que contenga al menos una imagen IRAP solo contendría imágenes IRAP. Por otro lado, si no se requiriera que las imágenes IRAP estuvieran alineadas, en una sola unidad de acceso, una imagen (por ejemplo, en una primera capa) puede ser una imagen IRAP, y otra imagen (por ejemplo, en una segunda capa) puede ser una imagen no IRAP. Tener tales imágenes IRAP no alineadas en un flujo de bits puede proporcionar algunas ventajas. Por ejemplo, en un flujo de bits de dos capas, si hay más imágenes IRAP en la capa base que en la capa de mejora, en aplicaciones de radiodifusión y multidifusión, se puede lograr un bajo retardo de sintonización y una alta eficiencia de codificación.

[0126] En algunos esquemas de codificación de vídeo, se puede usar un recuento de orden de imágenes (POC) para realizar un seguimiento del orden relativo en el que se muestran las imágenes descodificadas. Algunos de estos esquemas de codificación pueden hacer que los valores de POC se restablezcan (por ejemplo, se establezcan en cero o se establezcan en algún valor señalado en el flujo de bits) siempre que ciertos tipos de imágenes aparezcan en el flujo de bits. Por ejemplo, los valores POC de ciertas imágenes IRAP pueden restablecerse, lo que hace que los valores POC de otras imágenes que preceden a esas imágenes IRAP en el orden de descodificación también se restablezcan. Esto puede ser problemático cuando no es necesario que las imágenes IRAP se alineen en las diferentes capas. Por ejemplo, cuando una imagen ("picA") es una imagen IRAP y otra imagen ("picB") en la misma unidad de acceso no es una imagen IRAP, el valor POC de una imagen ("picC"), que se restablece debido a que picA sea una imagen IRAP en la capa que contiene picA puede ser diferente del valor POC de una imagen ("picD") que no se restablece, en la capa que contiene picB, donde picC y picD están en la misma unidad de acceso. Esto hace que picC y picD tengan diferentes valores de POC aunque pertenezcan a la misma unidad de acceso (por ejemplo, el mismo tiempo de salida). Por lo tanto, en este ejemplo, el proceso de obtención para obtener los valores POC de picC y picD se puede modificar para producir valores POC que sean coherentes con la definición de valores POC y unidades de acceso.

Recuento de orden de imágenes (POC)

[0127] Como se analizó anteriormente, el valor de un recuento de orden de imágenes (POC) (por ejemplo, PicOrderCntVal en HEVC) para una imagen codificada particular denota el orden relativo de la imagen codificada particular en el proceso de salida de imágenes con respecto a otras imágenes en la misma secuencia de vídeo codificado. En algunos modos de realización, POC incluye bits menos significativos (LSB) y bits más significativos (MSB); y el POC se puede obtener concatenando los MSB con los LSB. En otros modos de realización, el POC se puede obtener agregando el valor de MSB y el valor de LSB. El LSB puede ser señalado en la cabecera del fragmento, y el MSB puede ser calculado por el codificador o el descodificador basándose en el tipo de unidad NAL de la imagen actual y el MSB y el LSB de una o más imágenes previas en el orden de descodificación que (1) son imágenes no RASL o RADL, (2) son no desechables (por ejemplo, imágenes marcadas como "desechables", lo que indica que ninguna otra imagen depende de ellas, lo que permite eliminarlas para satisfacer las restricciones de ancho de banda), (3) no son imágenes de subcapa no de referencia (por ejemplo, imágenes que no se usan como referencia en otras imágenes en la misma subcapa temporal o en la misma capa), (4) tienen una identificación temporal (por ejemplo, un ID de subcapa temporal) igual a 0. Dichas imágenes descritas en (1)-(4) pueden referirse en el presente documento como imágenes de anclaje POC. De manera similar, las imágenes que tienen un valor de ID temporal mayor que 0, las imágenes RASL o RADL, las imágenes desechables o las imágenes de subcapa no de referencia pueden denominarse imágenes sin anclaje POC. Las imágenes de anclaje POC pueden incluir además imágenes que un codificador y/o un descodificador no pueden elegir eliminar del flujo de bits (por ejemplo, para satisfacer una restricción de ancho de banda). Las imágenes de anclaje POC pueden incluir además cualquier imagen distinta de los tipos de imágenes que un codificador y/o un descodificador pueden configurarse para eliminar del flujo de bits (por ejemplo, para satisfacer una restricción de ancho de banda). Las imágenes sin anclaje POC pueden incluir cualquier imagen que no sea una imagen de anclaje POC.

[0128] Cuando la imagen actual es (1) una imagen IRAP con NoRaslOutputFlag (por ejemplo, un indicador que indica que las imágenes RASL no se deben emitir si se establece en 1 e indica que las imágenes RASL se deben emitir si se establecen en 0) igual a 1 o (2) una imagen CRA que es la primera imagen del flujo de bits, se infiere que el valor de POC MSB es igual a 0. Tal como se ha descrito anteriormente, en un flujo de bits multicapa (por ejemplo, un flujo de bits SHVC o MV-HEVC con más de una capa), pueden existir unidades de acceso (AU) donde una o más imágenes son imágenes IRAP y una o más imágenes diferentes son imágenes no IRAP; y dichas AU se denominan "AU de IRAP no alineadas". Al descodificar flujos de bits que contienen AU de IRAP no alineadas,

es posible (y probable) que los POC obtenidos basándose en los valores POC LSB señalizados en el flujo de bits violarían el requisito de conformidad del flujo de bits de que todas las imágenes en una unidad de acceso deberían tener el mismo valor POC.

5 **[0129]** En algunos modos de realización, un indicador de restablecimiento de POC (por ejemplo, `poc_reset_flag`) se puede usar para restablecer el POC de las imágenes de tal manera que incluso cuando están presentes AU de IRAP no alineadas en el flujo de bits, el valor de POC de la imagen actual y las imágenes en la DPB se ajustan de tal manera que el POC de todas las imágenes en una AU sea el mismo.

10 **[0130]** En algunos modos de realización, en lugar de un solo indicador de restablecimiento de POC, se pueden usar dos indicadores: un indicador de restablecimiento de POC MSB (por ejemplo, `poc_msb_reset_flag`) y un indicador de restablecimiento de POC LSB (por ejemplo, `poc_lsb_reset_flag`). El primero (es decir, el `poc_msb_reset_flag`) restablece el MSB del POC, y el último (es decir, el `poc_lsb_reset_flag`) restablece el LSB del POC. Ambos indicadores se señalizan en la cabecera del fragmento.

15 **[0131]** Por ejemplo, si una imagen en particular tiene un valor de POC de 233, y el MSB y el LSB del POC constituyen 1 bit y 7 bits, respectivamente, el MSB sería "1" (por ejemplo, con un valor de 128) y el LSB sería "1101001" (por ejemplo, con un valor de 105). Por lo tanto, si solo se restablece el MSB del POC (por ejemplo, en respuesta al procesamiento de `poc_msb_reset_flag` que tiene un valor de 1), el valor del POC se convierte en 105, y si solo se restablece el LSB (por ejemplo, en respuesta al procesamiento del `poc_lsb_reset_flag` que tiene un valor de 1), el valor POC se convierte en 128. Si tanto el MSB como el LSB se restablecen (por ejemplo, en respuesta al procesamiento de `poc_msb_reset_flag` y `poc_lsb_reset_flag`, cada uno con un valor de 1), el valor de POC se convierte en 0.

25 **Restablecimiento de los valores de POC**

[0132] Con referencia a las **figuras 4-7**, se describirá la motivación para restablecer los valores de POC (por ejemplo, el LSB y el MSB) en las AU de IRAP no alineadas. Como se describió anteriormente, en algunos esquemas de codificación, ciertas restricciones de conformidad pueden especificar que el POC de todas las imágenes codificadas en una única AU debe ser el mismo. Sin los restablecimientos adecuados de los valores de POC, las AU de IRAP no alineadas en el flujo de bits pueden producir valores de POC que violan tales restricciones de conformidad.

35 **[0133]** La **figura 4** muestra un flujo de bits 400 multicapa que incluye una capa de mejora (EL) 410 y una capa base (BL) 420. La EL 410 incluye las imágenes EL 412-418, y la BL incluye las imágenes BL 422-428. El flujo de bits de múltiples capas 400 incluye además las unidades de acceso (AU) 430-460. La AU 430 incluye la imagen EL 412 y la imagen BL 422, la AU 440 incluye la imagen EL 414 y la imagen BL 424, la AU 450 incluye la imagen EL 416 y la imagen BL 426, y la AU 460 incluye la imagen EL 418 y la imagen BL 428. En el ejemplo de la **figura 4**, la imagen EL 414 es una imagen IDR, y la imagen BL 424 correspondiente en la AU 440 es una imagen final (por ejemplo, una imagen no IRAP), y en consecuencia, la AU 440 es una AU de IRAP no alineada. En algunos modos de realización, un restablecimiento de MSB se realiza en una imagen dada si la imagen es una imagen IDR que no está en la capa base. Dicha imagen IDR puede tener un valor POC LSB distinto de cero.

45 **[0134]** La **figura 5** muestra una tabla 500 que ilustra los valores de POC que pueden ser señalizados u obtenidos en relación con el flujo de bits multicapa 400 de la **figura 4**. Como se muestra en la **figura 5**, el MSB del POC en la EL 410 se restablece en la imagen EL 414, mientras que el MSB del POC en la BL 420 no se restablece. Por lo tanto, si no se realiza un restablecimiento en la BL 420 en la imagen BL 424 en la AU de IRAP no alineadas 440, los valores del POC de las imágenes BL y las imágenes EL en las AU 440-460 no coincidirían (es decir, serían equivalentes) según lo especificado por las restricciones de conformidad. Las diferencias en los valores de POC con y sin restablecimiento se resaltan en negrita en la **figura 5**.

55 **[0135]** La **figura 6** muestra un flujo de bits 600 multicapa que incluye una capa de mejora (EL) 610 y una capa base (BL) 620. La EL 610 incluye imágenes EL 612-618, y la BL incluye imágenes BL 622-628. El flujo de bits 600 de múltiples capas incluye además las unidades de acceso (AU) 630-660. La AU 630 incluye la imagen EL 612 y la imagen BL 622, la AU 640 incluye la imagen EL 614 y la imagen BL 624, la AU 650 incluye la imagen EL 616 y la imagen BL 626, y la AU 660 incluye la imagen EL 618 y la imagen BL 628. En el ejemplo de la **figura 6**, la imagen BL 624 es una imagen IDR, y la imagen EL 614 correspondiente en la AU 640 es una imagen final (por ejemplo, una imagen no IRAP), y en consecuencia, la AU 640 es una AU de IRAP no alineada. En algunos modos de realización, un restablecimiento de MSB y un restablecimiento de LSB se realizan para una imagen dada si la imagen es una imagen IDR que se encuentra en la capa base. Por ejemplo, el flujo de bits puede incluir una indicación de que el POC MSB y el POC LSB de tal imagen IDR de BL deben restablecerse. De forma alternativa, el descodificador puede realizar el restablecimiento del POC MSB y del POC LSB de tal imagen IDR de BL sin ninguna indicación en el flujo de bits de que se debe realizar un restablecimiento del POC.

65 **[0136]** La **figura 7** muestra una tabla 700 que ilustra los valores de POC que pueden ser señalizados u obtenidos en relación con el flujo de bits 600 multicapa de la **figura 6**. Como se muestra en la **figura 7**, el MSB y el LSB del

POC en la BL 620 se restablecen en la imagen BL 624, mientras que ni el MSB ni el LSB del POC en la EL 610 se restablecen. Por lo tanto, si no se realiza un restablecimiento del MSB y del LSB del POC en la EL 610 en la imagen EL 614 en la AU de IRAP no alineadas 640, los valores de POC de las imágenes BL y las imágenes EL en las AU 640-660 no coincidiría según lo especificado por las restricciones de conformidad. Las diferencias en los valores de POC con y sin restablecimiento se resaltan en negrita en la **figura 7**.

[0137] Los modos de realización descritos en el presente documento no están limitados a las configuraciones de flujo de bits de ejemplo ilustradas en las **figuras 4 y 6**, y las técnicas descritas en el presente documento pueden extenderse a cualquier flujo de bits multicapa que tenga cualquier número de capas, unidades de acceso e imágenes. Además, en los ejemplos ilustrados en las **figuras 4-7**, el LSB del POC se representa utilizando siete bits. Sin embargo, las técnicas descritas en el presente documento pueden extenderse a escenarios que tengan cualquier forma de representación de valor de POC.

Restablecimiento de imágenes anteriores y pérdida de imágenes restablecidas

[0138] Cuando se realiza un restablecimiento de MSB o un restablecimiento de LSB en una imagen en particular, otras imágenes en la misma capa que preceden a la imagen en particular en el orden de descodificación también se restablecen basándose en el restablecimiento realizado en la imagen en particular. Por ejemplo, en el ejemplo de la **figura 6**, la imagen EL 614 tiene un valor de POC de 241 (por ejemplo, LSB de "1110001" + MSB de "1", que es $113 + 128$). Cuando los restablecimientos de MSB y LSB se realizan en la imagen EL 614, el valor de POC de la imagen EL 614 se convierte en 0, y la imagen EL 612 en la EL 610 que precede a la imagen EL 614 en el orden de descodificación también se restablece basándose en el valor de POC original de 241 de la imagen EL 614. Por ejemplo, el nuevo valor de POC de la imagen EL 612 se calcula restando el valor de POC preestablecido de la imagen EL 614 (que es un valor de 241) del valor de POC preestablecido de la imagen EL 612, que es 240 (por ejemplo, LSB de "1110000" + MSB de "1", que es $112 + 128$). Por lo tanto, después del restablecimiento, el valor de POC de la imagen EL 612 se convierte en -1, de acuerdo con el hecho de que la imagen EL 612 debe emitirse antes que la imagen EL 614, donde un valor de POC menor denota una posición anterior en el orden de salida. Como se muestra en la **figura 7**, los valores de LSB señalizados para las AU posteriores 650 y 660 se ajustan en consecuencia (por ejemplo, a 1 y 2, respectivamente), con el supuesto de que el restablecimiento se realiza en la imagen EL 614.

[0139] Sin embargo, incluso si un restablecimiento de POC apropiado del MSB y/o el LSB descrito anteriormente se señala en el flujo de bits (por ejemplo, en la cabecera del fragmento) de modo que el descodificador pueda procesar la señal y realizar el restablecimiento de POC en consecuencia, si la señalización de la imagen de tal restablecimiento de POC se pierde durante la transmisión del flujo de bits o se elimina del flujo de bits para satisfacer las restricciones de ancho de banda, el restablecimiento de POC que se pretende realizar en la imagen en particular puede no realizarse correctamente.

[0140] Por ejemplo, en el ejemplo de la **figura 6**, si la imagen EL 614 no está disponible para el descodificador, el descodificador no sabría (es decir, no determinaría) restablecer el MSB y el LSB del POC en la EL 610 en la AU 640. En consecuencia, los valores de POC de cualquier imagen que precede a la imagen EL 614 no disponible en el orden de descodificación seguiría teniendo sus valores de POC preestablecidos originales, ya que el restablecimiento en la imagen EL 614 nunca tuvo lugar (es decir, no se realizó la operación de restablecimiento). Por otro lado, los valores de POC de las imágenes que siguen a la imagen EL 614 no disponible en el orden de descodificación se habrían determinado o señalizado como si realmente se hubiera realizado un restablecimiento (es decir, se realizó la operación de restablecimiento). Así, en el ejemplo de la **figura 7**, las imágenes EL 612, 616 y 618 tendrían valores de POC de 240, 1 y 2, respectivamente, lo que sería incorrecto dado que la imagen EL 612 precede a las imágenes EL 616 y 618 en el orden de salida. Por lo tanto, se desea un procedimiento de codificación que resulte en valores de POC correctos, incluso cuando la señalización de imagen del restablecimiento de POC no esté disponible.

Ejemplos e implementaciones

[0141] A continuación se describirán varios procedimientos que pueden usarse 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 pueden usarse para implementar uno o más de los procedimientos descritos en el presente documento también se proporcionan a continuación. Cuando ciertas partes de la especificación HEVC se reproducen para ilustrar las adiciones y eliminaciones que pueden incorporarse para implementar uno o más de los procedimientos descritos en el presente documento, tales adiciones se muestran en cursiva y tachadas, respectivamente.

Señalización de valores para la obtención de POC

[0142] En algunos modos de realización, un mensaje SEI que contiene información para la obtención del POC correcto se señala para una o más imágenes que siguen la imagen para la cual se restablecerá el POC MSB y/o el POC LSB. Por ejemplo, el mensaje SEI puede asociarse con una imagen, picA, que sigue a otra imagen, picB,

para la cual se restablecerá el POC MSB, el POC LSB o ambos. Por lo tanto, incluso cuando picB se pierde por completo, el mensaje SEI asociado con picA se puede usar para obtener los valores de POC correctos para otras imágenes en la misma capa.

5 **[0143]** En algunos modos de realización, la información para la obtención del POC correcto se señala en la cabecera del fragmento de una o más imágenes que siguen la imagen para la cual se restablecerá el POC MSB y/o el POC LSB. Por ejemplo, la información puede incluirse en la cabecera del fragmento de una imagen picA que sigue a otra imagen picB para la cual se restablecerán el POC MSB, el POC LSB o ambos. Por lo tanto, incluso cuando se pierde por completo PicB, la información incluida en la cabecera del fragmento de picA se puede usar para obtener los valores de POC correctos para otras imágenes en la misma capa.

Obtención del POC correcto basándose en los valores señalizados

15 **[0144]** La **figura 8** es un diagrama de flujo que ilustra un procedimiento 800 para codificar información de vídeo, de acuerdo a un modo de realización de la presente divulgación. Las etapas ilustradas en la **figura 8** pueden realizarse mediante un codificador (por ejemplo, el codificador de vídeo como se muestra en la **figura 2A** o la **figura 2B**), un descodificador (por ejemplo, el descodificador de vídeo como se muestra en la **figura 3A** o la **figura 3B**), o cualquier otro componente. Por conveniencia, el procedimiento 800 se describe como realizado por un codificador, que puede ser el codificador, el descodificador u otro componente.

20 **[0145]** El procedimiento 800 comienza en el bloque 801. En el bloque 805, el codificador procesa la información de obtención del POC asociada con una imagen. En algunos modos de realización, el procesamiento de la información de obtención del POC puede incluir señalar la información de obtención del POC en un flujo de bits. Como se describió anteriormente, la información de obtención del POC puede ser señalizada en la cabecera del fragmento de la imagen y/o señalizada como un mensaje SEI asociado con la imagen. En algunos modos de realización, el procesamiento de la información de obtención del POC puede incluir procesar la información de obtención del POC incluida en un flujo de bits. Por ejemplo, la información de obtención del POC puede incluir: un tipo de restablecimiento de POC que indique si el valor de POC de la imagen de restablecimiento de POC precedente (por ejemplo, una imagen en la que se realizará un restablecimiento de POC) en la misma capa se restablecerá mediante el restablecimiento de los bits más significativos (MSB) y los bits menos significativos (LSB) del valor del POC o mediante el restablecimiento solo de MSB del valor POC; un valor de restablecimiento de POC que indica el valor de POC de la imagen que se perdió o eliminó que también precede a la imagen con la que está asociada la información de obtención del POC; y un ID de restablecimiento de POC que identifica el restablecimiento de POC para el cual se proporciona la información de obtención del POC. Por ejemplo, el descodificador puede omitir un restablecimiento de POC señalado en conexión con una imagen particular si el restablecimiento del POC señalado tiene un valor de ID de restablecimiento del POC de 1 y ya se ha realizado otro restablecimiento del POC con un ID de restablecimiento del POC de 1.

40 **[0146]** En el bloque 810, el codificador determina el POC de otra imagen que precede a la imagen en el orden de descodificación. En el ejemplo mostrado en la **figura 7**, incluso si la imagen EL 614 que contiene la instrucción de restablecimiento del valor de POC se pierde o se elimina de otra manera, el valor de POC de la imagen EL 612 se restablecería correctamente usando la información de obtención del POC, por ejemplo, asociada con las imágenes EL 616 y/o 618. El procedimiento 800 termina en 815.

45 **[0147]** Como se analizó anteriormente, uno o más componentes del codificador de vídeo 20 de la **figura 2A**, el codificador de vídeo 23 de la **figura 2B**, el descodificador de vídeo 30 de la **figura 3A**, o el descodificador de vídeo 33 de la **figura 3B**.

50 **[0148]** (por ejemplo, la unidad de predicción entre capas 128 y/o la unidad de predicción entre capas 166) se puede usar para implementar cualquiera de las técnicas analizadas en la presente divulgación, como procesar la información de obtención de POC asociada con una o más imágenes y determinar el POC de otra imagen que precede a la una o más imágenes en el orden de descodificación.

Desactivación del restablecimiento de POC en imágenes sin anclaje POC

55 **[0149]** En algunos modos de realización, se puede aplicar una restricción de conformidad (por ejemplo, se puede determinar que una restricción de conformidad es aplicable y luego se cumple según el desempeño de la operación u operaciones), por ejemplo, mediante un codificador, al flujo de bits de tal manera que ni el valor de poc_lsb_reset_flag ni poc_msb_reset_flag se establecen igual a 1 cuando el segmento del fragmento pertenece a una imagen que no es una imagen de anclaje POC. Como se describió anteriormente, una imagen de este tipo puede ser una imagen de subcapa no de referencia, una imagen desechable, una imagen RASL, una imagen RADL o una imagen que tiene un ID temporal mayor que 0. Por ejemplo, la imagen de subcapa no de referencia puede referirse a una imagen que no se usa como referencia en otras imágenes de la capa temporal más alta. La imagen desechable puede referirse a una imagen que no se utiliza como referencia en ninguna otra imagen. Por ejemplo, tales imágenes desechables pueden estar marcadas como "desechables". Dichas imágenes desechables pueden ser eliminadas del flujo de bits por el codificador o el descodificador para satisfacer las restricciones de

ancho de banda. En algunos modos de realización, una imagen desechable incluye cualquier imagen que pueda eliminarse del flujo de bits por elección (por ejemplo, mediante el descodificador o alguna caja intermedia). Las imágenes RASL y RADL se refieren a las imágenes iniciales, y las imágenes RASL pueden no emitirse si el proceso de descodificación comienza en la imagen IRAP asociada con la imagen RASL. La imagen que tiene un ID temporal mayor que 0 puede ser una imagen que se puede eliminar del flujo de bits si la velocidad de trama se cambia a un valor suficientemente bajo. Por ejemplo, si un flujo de bits contiene tres subcapas temporales, las imágenes de las tres subcapas temporales pueden mostrarse para funcionar a 90 tramas por segundo (fps), las imágenes de las dos subcapas temporales inferiores pueden mostrarse para funcionar a 60 fps, y las imágenes de la subcapa temporal más baja pueden mostrarse para funcionar a 30 fps. Como se analizó anteriormente, las restricciones del flujo de bits u otras restricciones de rendimiento pueden hacer que una o más imágenes se eliminen o se retiren del flujo de bits (por ejemplo, un codificador puede evaluar tales restricciones y, basándose en esta evaluación, realizar operaciones de acuerdo con las restricciones de tal manera que se provoca que una o más imágenes se eliminen del flujo de bits o se retiren del flujo de bits), y en este ejemplo, las imágenes de la subcapa temporal más alta se pueden eliminar antes de eliminar las imágenes de la siguiente subcapa temporal más alta, y así sucesivamente. Por ejemplo, las imágenes en la subcapa temporal más baja pueden no eliminarse del flujo de bits hasta que se eliminen las imágenes en todas las demás subcapas temporales. Por lo tanto, es más probable que las imágenes con un ID temporal mayor que 0 (donde un ID temporal de 0 corresponde a la subcapa temporal más baja) se eliminen del flujo de bits.

[0150] Como se describe en el presente documento, estas imágenes (por ejemplo, una imagen de subcapa no de referencia, una imagen desechable, una imagen RASL, una imagen RADL, una imagen que tiene un ID temporal mayor que 0, y similares) pueden denominarse imágenes sin anclaje POC. En algunos modos de realización, debido a que es más probable que estas imágenes se eliminen del flujo de bits (por ejemplo, para satisfacer ciertas restricciones de ancho de banda), se puede introducir una restricción que especifica que estas imágenes no pueden activar un restablecimiento de POC para reducir la probabilidad de que una imagen de restablecimiento de POC pueda ser eliminada del flujo de bits. Por ejemplo, si no se permite que una imagen desechable active un restablecimiento de POC (por ejemplo, al señalar un restablecimiento de POC MSB, un restablecimiento de POC LSB, o ambos), incluso si se desecha la imagen desechable, la no disponibilidad de esa imagen desechable para el descodificador no causaría los problemas descritos anteriormente con respecto a los restablecimientos de POC.

[0151] En algunos modos de realización, el codificador puede determinar que un restablecimiento de POC debe ser señalado en relación con una imagen en particular, y posteriormente determinar que la imagen en particular es una imagen de subcapa no de referencia, una imagen desechable, una imagen RASL, una imagen RADL, una imagen que tiene un ID temporal mayor que 0, o una imagen que de otra manera es probable que se elimine del flujo de bits y, por lo tanto, se abstenga de señalar un restablecimiento de POC en la imagen en particular o señalar que no se debe realizar un restablecimiento de POC en la imagen en particular. En algunos modos de realización, el codificador puede determinar que un restablecimiento de POC debe ser señalado en relación con una imagen en particular y, posteriormente, evitar que la imagen en particular sea una imagen sin anclaje POC (por ejemplo, evitando que la imagen en particular tenga ciertos tipos de imagen). En algunos modos de realización, la determinación de si se debe realizar un restablecimiento de POC en la imagen en particular puede basarse, al menos en parte, en si la imagen en particular es una imagen de subcapa de no referencia, una imagen desechable, una imagen RASL, una imagen RADL, una imagen que tiene un ID temporal mayor que 0, o una imagen que probablemente de otro modo se eliminará del flujo de bits. En tales modos de realización, si la imagen en particular no es una imagen de anclaje POC, el codificador indica en el flujo de bits que el restablecimiento de POC no se debe realizar en la imagen en particular. De forma alternativa, el codificador puede simplemente no proporcionar ninguna indicación en el flujo de bits de que se deba realizar un restablecimiento de POC en la imagen en particular. De manera similar, si la imagen en particular es una imagen de anclaje POC, el codificador, si se determina que un restablecimiento de POC es necesario en la imagen en particular, indica en el flujo de bits que el restablecimiento de POC se realizará en la imagen en particular. De forma alternativa, el codificador puede simplemente no proporcionar ninguna indicación en el flujo de bits de que el restablecimiento de POC no se va a realizar o de que el restablecimiento de POC no debe realizarse en la imagen en particular.

Desactivación del restablecimiento de POC en imágenes sin anclaje POC

[0152] La **figura 9** es un diagrama de flujo que ilustra un procedimiento 900 para codificar información de vídeo, de acuerdo a un modo de realización de la presente divulgación. Las etapas ilustradas en la **figura 9** puede realizarse mediante un codificador (por ejemplo, el codificador de vídeo como se muestra en la **figura 2A** o la **figura 2B**), un descodificador (por ejemplo, el descodificador de vídeo como se muestra en la **figura 3A** o la **figura 3B**), o cualquier otro componente. Por conveniencia, el procedimiento 900 se describe como realizado por un codificador, que puede ser el codificador, el descodificador u otro componente.

[0153] El procedimiento 900 comienza en el bloque 901. En el bloque 905, el codificador determina si una imagen es una imagen de anclaje POC. Por ejemplo, las imágenes de anclaje POC pueden incluir cualquier imagen que: (1) no sean imágenes RASL o RADL, (2) sean no desechables (por ejemplo, imágenes marcadas como "desechables", lo que indica que ninguna otra imagen depende de ellas, lo que permite que se eliminen para satisfacer las restricciones de ancho de banda), (3) no sean imágenes de subcapa de no referencia (por ejemplo,

las imágenes que no se usan como referencia en otras imágenes en las capas temporales superiores), (4) tienen un ID temporal (por ejemplo, un ID de subcapa temporal) igual a 0, y/o (5) cualquier otra imagen que de lo contrario es probable que se elimine del flujo de bits. Si el codificador determina que la imagen no es una imagen de anclaje POC, el procedimiento 900 pasa a 910. Por otro lado, si el codificador determina que la imagen es una imagen de anclaje POC, el procedimiento 900 continúa hasta 915.

[0154] En el bloque 910, el codificador señala para la imagen que el restablecimiento de POC no se realizará en la imagen. Por ejemplo, el codificador puede señalar uno o más indicadores que indican que ni el restablecimiento de POC LSB ni el restablecimiento de POC MSB se deben realizar en relación con la imagen. En algunos modos de realización, el codificador puede no señalar o proporcionar de otro modo ninguna indicación en el flujo de bits de que se debe realizar un restablecimiento de POC en la imagen. Por ejemplo, durante el proceso de descodificación, si no se proporciona ninguna señal o indicación que indique que se debe realizar un restablecimiento de POC en el flujo de bits, el descodificador puede no realizar un restablecimiento de POC en esa imagen.

[0155] En el bloque 915, el codificador señala un restablecimiento de POC para la imagen. Por ejemplo, el codificador puede señalar uno o más indicadores en el flujo de bits que indican que se debe realizar un restablecimiento de POC LSB, un restablecimiento de POC MSB o ambos. En algunos modos de realización, el codificador puede no señalar o proporcionar de otro modo ninguna indicación en el flujo de bits de que no se debe realizar un restablecimiento de POC en la imagen. Por ejemplo, durante el proceso de descodificación, el descodificador puede inferir o determinar a partir de otras señales o indicaciones en el flujo de bits que se realizará un restablecimiento de POC, y que si no se proporciona ninguna señal o indicación adicional que deshabilite el restablecimiento de POC en el flujo de bits, el descodificador debe realizar el restablecimiento de POC como inferido o determinado. El procedimiento 900 termina en 920.

[0156] Como se analizó anteriormente, uno o más componentes del codificador de vídeo 20 de la **figura 2A**, el codificador de vídeo 23 de la **figura 2B**, el descodificador de vídeo 30 de la **figura 3A**, o el descodificador de vídeo 33 de la **figura 3B** (por ejemplo, la unidad de predicción entre capas 128 y/o la unidad de predicción entre capas 166) pueden usarse para implementar cualquiera de las técnicas analizadas en la presente divulgación, como determinar si una imagen es una imagen de anclaje POC, permitir un restablecimiento de POC, deshabilitar un restablecimiento de POC, proporcionar una indicación en el flujo de bits de que se debe realizar un restablecimiento de POC y proporcionar una indicación en el flujo de bits de que no se debe realizar un restablecimiento de POC.

[0157] En el procedimiento 900, uno o más de los bloques mostrados en la **figura 9** pueden eliminarse (por ejemplo, no realizarse) y/o puede conmutarse el orden en que se realiza el procedimiento. Por ejemplo, aunque el bloque 910 se muestra en la **figura 9**, el bloque 910 puede eliminarse, y el procedimiento 900 puede finalizar sin realizar ninguna operación adicional si el codificador determina que la imagen no es una imagen de anclaje POC. De forma alternativa, el bloque 915 puede eliminarse, y el procedimiento 900 puede finalizar sin realizar ninguna operación adicional si el codificador determina que la imagen es una imagen de anclaje POC. Así pues, los modos de realización de la presente divulgación no están limitados a, o por, el ejemplo mostrado en la **figura 9**.

Imágenes IRAP en AU de IRAP no alineadas

[0158] En algunos modos de realización, se puede aplicar una restricción de conformidad al flujo de bits de modo que cuando una unidad de acceso contenga al menos una imagen que sea una imagen IRAP con NoRasOutputFlag igual a 1, se realizará un restablecimiento de POC MSB (por ejemplo, MSB del POC) para todas las imágenes en la unidad de acceso que no sean imágenes IRAP. En tales modos de realización, poc_msb_reset_flag asociado con las imágenes no IRAP se puede establecer en 1 (por ejemplo, lo que indica que se debe realizar un restablecimiento de POC MSB en dichas imágenes no IRAP). Por ejemplo, si la Imagen A es una imagen IRAP en una unidad de acceso que sigue inmediatamente a un punto de empalme (por ejemplo, indicado por el valor NoRasOutputFlag de 1), y la Imagen B que está en la misma unidad de acceso que la Imagen A es una imagen no IRAP, se puede señalar un restablecimiento de POC MSB en el flujo de bits para la Imagen B.

[0159] La **figura 10** es un diagrama de flujo que ilustra un procedimiento 1000 para codificar información de vídeo, de acuerdo a un modo de realización de la presente divulgación. Las etapas ilustradas en la **figura 10** pueden realizarse mediante un codificador (por ejemplo, el codificador de vídeo como se muestra en la **figura 2A** o la **figura 2B**), un descodificador (por ejemplo, el descodificador de vídeo como se muestra en la **figura 3A** o la **figura 3B**), o cualquier otro componente. Por conveniencia, el procedimiento 1000 se describe como realizado por un codificador, que puede ser el codificador, el descodificador u otro componente.

[0160] El procedimiento 1000 comienza en el bloque 1001. En el bloque 1005, el codificador determina si una imagen es una imagen IRAP. Como se describió anteriormente, en algunos modos de realización, una imagen IRAP puede ser una imagen IDR, una imagen CRA o una imagen BLA. En algunos modos de realización, el codificador puede determinar adicionalmente, basándose en la información incluida en el flujo de bits, si la imagen está en una unidad de acceso que sigue inmediatamente a un punto de empalme. En algunos modos de

realización, el codificador puede determinar además, en lugar de determinar si la imagen está en una unidad de acceso que sigue inmediatamente a un punto de empalme, si las imágenes que preceden a la imagen en el orden de descodificación deben emitirse. Por ejemplo, si la imagen se encuentra en una unidad de acceso que sigue inmediatamente a un punto de empalme o si las imágenes que preceden a la imagen en el orden de descodificación deben emitirse pueden indicarse mediante una o más variables que están señalizadas o indicadas en el flujo de bits u obtenidas de otra información. disponible para el codificador (por ejemplo, NoRasIOutputFlag). Por ejemplo, para imágenes IDR e imágenes CRA, NoRasIOutputFlag puede obtenerse de otra información incluida en el flujo de bits. Para imágenes BLA, la presencia de tales imágenes BLA puede indicar al descodificador que las imágenes BLA siguen inmediatamente un punto de empalme. Si el codificador determina que la imagen es una imagen IRAP, el procedimiento 1000 continúa con el bloque 1010. De otro modo, el procedimiento 1000 termina en 1015.

[0161] En el bloque 1010, el codificador habilita un restablecimiento de POC MSB para todas las demás imágenes no IRAP en la unidad de acceso. En algunos modos de realización, el codificador habilita un restablecimiento de POC MSB para todas las demás imágenes no IRAP en la unidad de acceso que siguen inmediatamente un punto de empalme en orden de descodificación. Por ejemplo, el codificador puede señalar un indicador de restablecimiento de POC MSB (por ejemplo, poc_msb_reset_flag) que tiene un valor de 1, lo que indica que se debe realizar un restablecimiento de POC MSB para cada una de las imágenes no IRAP. El procedimiento 1000 termina en 1015.

[0162] Como se analizó anteriormente, uno o más componentes del codificador de vídeo 20 de la **figura 2A**, el codificador de vídeo 23 de la **figura 2B**, el descodificador de vídeo 30 de la **figura 3A**, o el descodificador de vídeo 33 de la **figura 3B** (por ejemplo, la unidad de predicción entre capas 128 y/o la unidad de predicción entre capas 166) se puede usar para implementar cualquiera de las técnicas analizadas en la presente divulgación, como determinar si una imagen es una imagen IRAP y habilitar un restablecimiento de POC MSB para todas las demás imágenes no IRAP en la unidad de acceso.

[0163] En el procedimiento 1000, uno o más de los bloques mostrados en la **figura 10** pueden eliminarse (por ejemplo, no realizarse) y/o puede conmutarse el orden en que se realiza el procedimiento. Los modos de realización de la presente divulgación no están limitados a, o por, el ejemplo mostrado en la **figura 10**.

Imágenes IDR de capa base en AU de IRAP no alineadas

[0164] En algunos modos de realización, se puede aplicar una restricción de conformidad al flujo de bits, de manera que cuando una unidad de acceso A contiene una imagen de capa base que es una imagen IDR, se realizará un restablecimiento de un POC LSB (por ejemplo, LSB del POC) para todas las imágenes de la capa de mejora en la unidad de acceso A que no son imágenes IDR o que tienen un valor de POC LSB distinto de cero señalizado en el flujo de bits. En tales modos de realización, el indicador poc_lsb_reset asociado con las imágenes EL (por ejemplo, que indica que se debe realizar un restablecimiento de POC LSB en dichas imágenes EL). Por ejemplo, si la Imagen A en la capa base es una imagen IDR, y la Imagen B que se encuentra en la misma unidad de acceso que la Imagen A no es una imagen IDR, se puede señalar un restablecimiento de POC LSB en el flujo de bits para la Imagen B. En otro ejemplo, si la Imagen A en la capa base es una imagen IDR, y la Imagen C en la misma unidad de acceso que la Imagen A tiene un valor POC LSB de 0 señalizado en el flujo de bits, es posible que no sea necesario señalar un restablecimiento de POC LSB en el flujo de bits para la imagen C.

[0165] La **figura 11** es un diagrama de flujo que ilustra un procedimiento 1100 para codificar información de vídeo, de acuerdo con un modo de realización de la presente divulgación. Las etapas ilustradas en la **figura 11** pueden realizarse mediante un codificador (por ejemplo, el codificador de vídeo como se muestra en la **figura 2A** o la **figura 2B**), un descodificador (por ejemplo, el descodificador de vídeo como se muestra en la **figura 3A** o la **figura 3B**), o cualquier otro componente. Por conveniencia, el procedimiento 1100 se describe como realizado por un codificador, que puede ser el codificador, el descodificador u otro componente.

[0166] El procedimiento 1100 comienza en el bloque 1101. En el bloque 1105, el codificador determina si una imagen es una imagen IDR de capa base. En algunos modos de realización, el POC de una imagen IDR de BL se restablece automáticamente a 0. Si el codificador determina que la imagen es una imagen IDR de BL, el procedimiento 1100 procede al bloque 1110. El procedimiento 1100 termina en 1115.

[0167] En el bloque 1110, el codificador habilita un restablecimiento de POC LSB para todas las demás imágenes no IDR en la unidad de acceso. Por ejemplo, el codificador puede señalar un indicador de restablecimiento de POC LSB (por ejemplo, poc_lsb_reset_flag) que tiene un valor de 1, lo que indica que se debe realizar un restablecimiento de POC LSB para cada una de las imágenes no IDR en la misma unidad de acceso que la Imagen IDR de BL. El procedimiento 1100 termina en 1115.

[0168] Como se analizó anteriormente, uno o más componentes del codificador de vídeo 20 de la **figura 2A**, el codificador de vídeo 23 de la **figura 2B**, el descodificador de vídeo 30 de la **figura 3A**, o el descodificador de vídeo 33 de la **figura 3B** (por ejemplo, la unidad de predicción entre capas 128 y/o la unidad de predicción entre capas 166) se puede utilizar para implementar cualquiera de las técnicas analizadas en la presente divulgación, como

determinar si una imagen es una imagen IDR de BL y habilitar un restablecimiento de POC LSB para todas las demás imágenes no IDR en la unidad de acceso.

5 [0169] En el procedimiento 1100, uno o más de los bloques mostrados en la **figura 11** pueden eliminarse (por ejemplo, no realizarse) y/o puede conmutarse el orden en que se realiza el procedimiento. Los modos de realización de la presente divulgación no están limitados a, o por, el ejemplo mostrado en la **figura 11**.

Señalización de restablecimiento de copia de seguridad en imágenes posteriores

10 [0170] En algunos modos de realización, para cada imagen picA que restablece su valor de POC MSB para alinear el POC obtenido (por ejemplo, PicOrderCntVal) con otra imagen que es una imagen IRAP con NoRaslOutputFlag igual a 1 y que está en la misma unidad de acceso que la picA, se puede proporcionar una indicación en el flujo de bits en asociación con una o más imágenes en la misma capa que picA que siguen a picA en el orden de descodificación de que se debe realizar un restablecimiento de POC MSB con una o más imágenes.
15 Por ejemplo, poc_msb_reset_flag con un valor de 1 puede ser señalado para cada una de las una o más imágenes.

20 [0171] La **figura 12** es un diagrama de flujo que ilustra un procedimiento 1200 para codificar información de vídeo, de acuerdo con un modo de realización de la presente divulgación. Las etapas ilustradas en la **figura 12** pueden realizarse mediante un codificador (por ejemplo, el codificador de vídeo como se muestra en la **figura 2A** o la **figura 2B**), un descodificador (por ejemplo, el descodificador de vídeo como se muestra en la **figura 3A** o la **figura 3B**), o cualquier otro componente. Por conveniencia, el procedimiento 1200 se describe como realizado por un codificador, que puede ser el codificador, el descodificador u otro componente.

25 [0172] El procedimiento 1200 comienza en el bloque 1201. En el bloque 1205, el codificador determina si se debe realizar un restablecimiento de POC MSB en una imagen en particular. Como se describió anteriormente, en algunos modos de realización, se puede realizar un restablecimiento de POC MSB para evitar que las imágenes en diferentes capas tengan valores de POC incompatibles en una unidad de acceso de IRAP no alineadas. En algunos modos de realización, el codificador determina además si la imagen particular es una imagen no IRAP en una unidad de acceso de IRAP no alineadas. Si el codificador determina que se debe realizar un restablecimiento de POC MSB en la imagen en particular, el procedimiento 1200 continúa con el bloque 1210. De otro modo, el procedimiento 1200 termina en 1215.

35 [0173] En el bloque 1210, el codificador habilita un restablecimiento de POC MSB para una o más imágenes que siguen la imagen en particular en el orden de descodificación. En algunos modos de realización, la una o más imágenes pueden estar en la misma capa que la imagen en particular. Por ejemplo, el codificador puede señalar un indicador de restablecimiento de POC MSB (por ejemplo, poc_msb_reset_flag) que tiene un valor de 1 para la imagen que sigue inmediatamente a la imagen en particular en el orden de descodificación, lo que indica que se debe realizar un restablecimiento de POC MSB para la imagen que sigue inmediatamente a la imagen en particular en el orden de descodificación. Como se describió anteriormente, si la imagen en particular que tiene una indicación de que se debe realizar un restablecimiento de POC MSB se pierde, se restablece una copia del POC MSB en la imagen que sigue inmediatamente a la imagen en particular en el orden de descodificación basándose en la indicación asociada con la imagen que sigue inmediatamente a la imagen en particular en el orden de descodificación. En algunos modos de realización, el codificador puede proporcionar además una indicación o variable que se puede usar para que el restablecimiento de POC MSB no se realice más de una vez. Dicha indicación o variable (por ejemplo, un ID de restablecimiento de POC) puede ayudar a determinar si se ha realizado el restablecimiento de POC MSB. En algunos modos de realización, el codificador habilita el restablecimiento de POC MSB para una o más imágenes solo si la imagen en particular es una imagen no IRAP en una unidad de acceso de IRAP no alineadas. El procedimiento 1200 termina en 1215.

50 [0174] Como se analizó anteriormente, uno o más componentes del codificador de vídeo 20 de la **figura 2A**, el codificador de vídeo 23 de la **figura 2B**, el descodificador de vídeo 30 de la **figura 3A**, o el descodificador de vídeo 33 de la **figura 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, como determinar si se debe realizar un restablecimiento de POC MSB en una imagen en particular, y habilitar un restablecimiento de POC MSB para una o más imágenes que siguen a la imagen en particular en el orden de descodificación.

60 [0175] En el procedimiento 1200, uno o más de los bloques mostrados en la **figura 12** pueden eliminarse (por ejemplo, no realizarse) y/o puede conmutarse el orden en que se realiza el procedimiento. Así pues, los modos de realización de la presente divulgación no están limitados a, o por, el ejemplo mostrado en la **figura 12**.

Señalización de valores de POC de imágenes anteriores

65 [0176] En algunos modos de realización, para la imagen picA que restablece su valor de POC para alinear el PicOrderCntVal obtenido con una imagen IDR que tiene nuh_layer_id igual a 0 y que está en la misma unidad de

acceso que picA, el valor de PicOrderCntVal de picA antes del restablecimiento de POC es señalizado para una o más imágenes que siguen a picA en el orden de descodificación y que tienen el mismo nuh_layer_id que picA.

Implementación de ejemplo: Sintaxis de la cabecera de segmento del fragmento

5

[0177] La siguiente sintaxis de cabecera de segmento del fragmento de ejemplo se puede usar para implementar uno o más de los modos de realización descritos en el presente documento.

Tabla 1. Sintaxis de cabecera de segmento del fragmento de ejemplo

10

	Descriptor
slice_segment_header() {	
first_slice_segment_in_pic_flag	u(1)
if(nal_unit_type >= BLA_W_LP && nal_unit_type <= RSV_IRAP_VCL23)	
no_output_of_prior_pics_flag	u(1)
slice_pic_parameter_set_id	ue(v)
if(! first_slice_segment_in_pic_flag) {	
if(dependent_slice_segments_enabled_flag)	
dependent_slice_segment_flag	u(1)
slice_segment_address	u(v)
}	
if(!dependent_slice_segment_flag) {	
i = 0	
if(num_extra_slice_header_bits > i) {	
i++	
poc_msb_reset_flag	u(1)
}	
if(num_extra_slice_header_bits > i) {	
i++	
poc_lsb_reset_flag	u(1)
}	
if(num_extra_slice_header_bits > i) {	
i++	
discardable_flag	u(1)
}	
for(; i < num_extra_slice_header_bits; i++)	
slice_reserved_flag[i]	u(1)
slice_type	ue(v)
if(output_flag_present_flag)	
pic_output_flag	u(1)
if(separate_colour_plane_flag == 1)	
colour_plane_id	u(2)
if(nuh_layer_id > 0 (nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP)) {	
slice_pic_order_cnt_lsb	u(v)
...	

Implementación de ejemplo: Semántica de la cabecera de segmento del fragmento

[0178] La siguiente semántica de ejemplo se puede usar para implementar uno o más de los modos de realización descritos en el presente documento. Los cambios al lenguaje existente en la especificación HEVC se muestran en *cursiva*.

[0179] **poc_msb_reset_flag** igual a 1 especifica que el valor de MSB del recuento del orden de imágenes obtenido para la imagen actual es igual a 0. **poc_msb_reset_flag** igual a 0 especifica que el valor de MSB del recuento del orden de imágenes obtenido para la imagen actual puede o no ser igual a 0.

[0180] *Cuando la imagen actual no es una imagen IRAP con NoRaslOutputFlag igual a 1 y al menos una imagen en la unidad de acceso actual es una imagen IRAP con NoRaslOutputFlag igual a 1, poc_msb_reset_flag estará presente y el valor será igual a 1.*

[0181] Cuando no está presente, se infiere que el valor de poc_msb_reset_flag es igual a 0. **poc_lsb_reset_flag** igual a 1 especifica que el recuento del orden de imágenes obtenido para la imagen actual es igual a 0, poc_lsb_reset_flag igual a 0 especifica que el recuento del orden de imágenes obtenido para la imagen actual puede o no ser igual a 0.

[0182] *Cuando la imagen actual no es una imagen IDR o slice_pic_order_cnt_lsb no es igual a 0, y la imagen con nuh_layer_id igual a 0 en la unidad de acceso actual es una imagen IDR, poc_lsb_reset_flag estará presente y el valor será igual a 1.*

[0183] Cuando no está presente, se infiere que el valor de poc_lsb_reset_flag es igual a 0.

[0184] *Cuando el valor de poc_msb_reset_flag es igual a 0, el valor de poc_lsb_reset_flag será igual a 0.*

[0185] *Es un requisito de conformidad con el flujo de bits que, cuando hay una imagen IRAP con NoRaslOutputFlag igual a 1 en la unidad de acceso actual, la imagen actual debe tener discardable_flag igual a 0 y TemporalId mayor que 0, y no debe ser una imagen de subcapa de no referencia, una imagen RASL o una imagen RADL.*

[0186] De forma alternativa, las siguientes restricciones podrían agregarse a la semántica de poc_msb_reset_flag y poc_lsb_reset_flag:

Es un requisito de conformidad del flujo de bits que para los fragmentos que tienen un valor de nal_unit_type menor que 16, y que tienen nal_unit_type % 2 = 0, el valor de ambas marcas de restablecimiento de poc_lsb_reset_flag y poc_msb_reset_flag será igual a 0.

Es un requisito de conformidad del flujo de bits que cuando el valor de poc_lsb_reset_flag o poc_msb_reset_flag, o ambos, sea igual a 1, el valor del indicador desechable, cuando esté presente, sea igual a 0.

Es un requisito de conformidad del flujo de bits que cuando una unidad de acceso contiene una imagen que es una imagen IRAP con NoRaslOutputFlag igual a 1, entonces se aplican las siguientes condiciones: (1) si la imagen con nuh_layer_id igual a 0 es una imagen IDR, el valor de poc_lsb_reset_flag y poc_msb_reset_flag se establecerá igual a 1 para todas las imágenes en esa unidad de acceso que tengan nuh_layer_id distinto de 0; y (2) de lo contrario, el valor de poc_lsb_reset_flag se establecerá igual a 1 y el valor de poc_msb_reset_flag se establecerá igual a 1 para todas las imágenes en esa unidad de acceso que no sean imágenes IRAP con NoRaslOutputFlag igual a 1.

Implementación de ejemplo: Proceso de decodificación para POC y el conjunto de imágenes de referencia

[0187] A continuación se describe una obtención de ejemplo del POC para cada fragmento. Cuando el valor de poc_lsb_reset_flag o poc_msb_reset_flag se establece en 1, el POC de la imagen actual y todas las imágenes en la DPB que están marcadas como "usadas como referencia" o que son necesarias para la salida se disminuyen.

Proceso de decodificación para el recuento del orden de imágenes

[0188] *La salida de este proceso es PicOrderCntVal, el recuento del orden de imágenes de la imagen actual.*

[0189] *Los recuentos del orden de imágenes se usan para identificar imágenes, para obtener parámetros de movimiento en modo de combinación y predicción de vectores de movimiento, y para la verificación de conformidad del decodificador.*

[0190] Cada imagen codificada está asociada con una variable de recuento del orden de imágenes, denotada como *PicOrderCntVal*.

5 **[0191]** Cuando la imagen actual no es una imagen IRAP con *NoRasOutputFlag* igual a 1, las variables *prevPicOrderCntLsb* y *prevPicOrderCntMsb* se obtienen de la siguiente manera:

10 **[0192]** Supongamos que *prevTid0Pic* es la imagen anterior en el orden de decodificación que tiene *Tcnporalld* igual a 0 y *nuh_layer_id* igual a *nuh_layer_id* de la imagen actual y que no es una imagen RASL, una imagen RADL o una imagen de subcapa de no referencia, y supongamos que *prevPicOrderCnt* igual a *PicOrderCntVal* de *prevTid0Pic*.

La variable *prevPicOrderCntLsb* se establece igual a $prevPicOrderCnt \& (MaxPicOrderCntLsb - 1)$.

15 La variable *prevPicOrderCntMsb* se establece igual a $prevPicOrderCnt - prevPicOrderCntLsb$.

[0193] La variable *PicOrderCntMsb* de la imagen actual se obtiene de la forma siguiente:

Si la imagen actual es una imagen IRAP con *NoRasOutputFlag* igual a 1, *PicOrderCntVal* se establece igual a 0.

De lo contrario, *PicOrderCntMsb* se obtiene de la forma siguiente:

```

if( ( slice_pic_order_cnt_lsb < prevPicOrderCntLsb ) &&
    ( ( prevPicOrderCntLsb - slice_pic_order_cnt_lsb ) >= (
MaxPicOrderCntLsb / 2 ) ) )
    PicOrderCntMsb = prevPicOrderCntMsb + MaxPicOrderCntLsb
else if( ( slice_pic_order_cnt_lsb > prevPicOrderCntLsb ) &&
    ( ( slice_pic_order_cnt_lsb - prevPicOrderCntLsb ) > (
MaxPicOrderCntLsb / 2 ) ) )
    PicOrderCntMsb = prevPicOrderCntMsb - MaxPicOrderCntLsb
else
    PicOrderCntMsb = prevPicOrderCntMsb

```

PicOrderCntVal se obtiene de la siguiente manera:

$$PicOrderCntVal = (poc_msb_reset_flag ? 0 : PicOrderCntMsb) + (poc_lsb_reset_flag ? 0 : slice_pic_order_cnt_lsb)$$

20 **[0194]** Se debe tener en cuenta que todas las imágenes IDR que tienen *nuh_layer_id* igual a 0 tendrán *PicOrderCntVal* igual a 0, ya que se infiere que *slice_pic_order_cnt_lsb* es 0 para las imágenes IDR y *prevPicOrderCntLsb* y *prevPicOrderCntMsb* se han establecido ambos en 0.

25 **[0195]** Cuando *poc_msbreset_flag* es igual a 1, *PicOrderCntVal* de cada imagen que está en la DPB y pertenece a la misma capa que la imagen actual se disminuye en *PicOrderCntVal*.

[0196] Cuando *poc_lsb_reset_flag* es igual a 1, *PicOrderCntVal* de cada imagen que está en la DPB y pertenece a la misma capa que la imagen actual se disminuye en *slice_pic_order_cnt_lsb*.

30 **[0197]** El valor de *PicOrderCntVal* estará dentro del intervalo de -231 a 231 - 1, inclusive. En una CVS, los valores de *PicOrderCntVal* para dos imágenes codificadas cualesquiera en la misma capa no serán iguales.

[0198] La función *PicOrderCnt(picX)* se especifica de la siguiente manera:

35 $PicOrderCnt(picX) = PicOrderCntVal$ de la imagen *picX*

[0199] La función *DiffPicOrderCnt(picA, picB)* se especifica de la siguiente manera:

40 $DiffPicOrderCnt(picA, picB) = PicOrderCnt(picA) - PicOrderCnt(picB)$

[0200] El flujo de bits no debe contener datos que den como resultado valores de $DiffPicOrderCnt(picA, picB)$ usados en el proceso de decodificación que no estén en el intervalo de -215 a $215 - 1$, inclusive.

5 **[0201]** Cabe destacar que si X es la imagen actual e Y y Z son otras dos imágenes en la misma secuencia, se considera que Y y Z están en la misma dirección de orden de salida desde X cuando tanto $DiffPicOrderCnt(X, Y)$ como $DiffPicOrderCnt(X, Z)$ son positivos o ambos son negativos.

Proceso de decodificación para el conjunto de imágenes de referencia

10 **[0202]** El proceso de decodificación para el conjunto de imágenes de referencia es idéntico al proceso de decodificación definido en MV-HEVC WD5.

Implementación de ejemplo: Sintaxis general de la carga útil de SEI

15 **[0203]** La siguiente sintaxis de carga útil de SEI de ejemplo puede usarse para implementar uno o más de los modos de realización descritos en el presente documento. En el siguiente ejemplo, "XXX" puede reemplazarse con cualquier valor que represente el tipo de carga útil que se puede utilizar en relación con la sintaxis de ejemplo. Por ejemplo, "XXX" puede reemplazarse con cualquier valor entre 1 y 255 que no haya sido usado por otro mensaje SEI. En otro ejemplo, el valor de "XXX" no está limitado a 255, y tiene un valor más alto. Los cambios al lenguaje existente en la especificación HEVC se muestran en *cursiva*.

20

Tabla 2. Sintaxis de carga útil de SEI de ejemplo

	Descriptor
sei_payload(payloadType, payloadSize) {	
if(nal_unit_type == PREFIX_SEI_NUT)	
if(payloadType == 0)	
...	
else if(payloadType == XXX)	
three_dimensional_reference_displays_info(payloadSize)	
<i>else if(payloadType == XXX)</i>	
<i>poc_reset_info(payloadSize)</i>	
...	
else	
reserved_sei_message(payloadSize)	
else /* nal_unit_type == SUFFIX_SEI_NUT */	
if(payloadType == 3)	
filler_payload(payloadSize)	
...	
else	
reserved_sei_message(payloadSize)	
if(more_data_in_payload()) {	
if(payload_extension_present())	
reserved_payload_extension_data	u(v)
payload_bit_equal_to_one /* igual a 1 */	f(1)
while(!byte_aligned())	
payload_bit_equal_to_zero /* igual a 0 */	f(1)
}	
}	
}	

25 **Implementación de ejemplo: Sintaxis de mensajes SEI de información de restablecimiento de POC**

[0204] La siguiente sintaxis de información de restablecimiento de POC de ejemplo puede usarse para implementar uno o más de los modos de realización descritos en el presente documento. Los cambios al lenguaje existente en la especificación HEVC se muestran en *cursiva*.

5 **Tabla 3. Sintaxis de información de restablecimiento de POC de ejemplo**

<i>poc_reset_info(payloadSize) {</i>	Descriptor
<i>poc_reset_type_flag</i>	u(1)
<i>poc_reset_value</i>	u(32)
<i>poc_reset_id</i>	u(7)
<i>}</i>	

[0205] En algunos modos de realización, *poc_reset_value*, *poc_reset_id*, o ambos, se codifican utilizando códigos exponencial-Golomb (por ejemplo, codificación ue(v)).

10 **Implementación de ejemplo: Semántica de mensajes SEI de información de restablecimiento de POC**

[0206] La siguiente semántica de información de restablecimiento de POC de ejemplo se puede usar para implementar uno o más de los modos de realización descritos en el presente documento: *"El mensaje SEI de información de restablecimiento de POC proporciona información que permite la obtención del POC correcto para la imagen asociada, incluso cuando la imagen anterior en el orden de descodificación en la misma capa que la imagen asociada y que tiene poc_lsb_reset_flag o poc_msb_reset_flag igual a 1 se pierde. La información contenida en el mensaje SEI también se puede usar para obtener los valores POC de otras imágenes en la DPB que están en la misma capa que la imagen asociada. La imagen de restablecimiento de POC se define como una imagen que tiene el valor de poc_msb_reset_flag o de poc_lsb_reset_flag, o ambos, igual a 1. La imagen de restablecimiento de POC asociada se refiere a la imagen anterior en el orden de descodificación en la misma capa que la imagen asociada y que tiene poc_lsb_reset_flag o poc_msb_reset_flag igual a 1. La imagen asociada de un mensaje SEI de información de restablecimiento de POC se refiere a la imagen que está asociada con el mensaje SEI. Un mensaje SEI de información de restablecimiento de POC no anidado se asocia con la imagen para la cual la primera unidad NAL de VCL en el orden de descodificación es la unidad NAL de VCL asociada de la unidad NAL de SEI que contiene el mensaje SEI de información de restablecimiento de POC no anidado. nuh_layer_id de la unidad NAL de SEI que contiene un mensaje SEI de información de restablecimiento de POC no anidado será igual a nuh_layer_id de la imagen asociada."*

[0207] De forma alternativa, la asociación del mensaje SEI se puede definir de la siguiente manera: *"Un mensaje SEI de información de restablecimiento de POC no anidado se asocia con la imagen picA en la siguiente unidad de acceso en la descodificación, donde picA tiene el mismo valor de nuh_layer_id que la unidad NAL de SEI que contiene el mensaje SEI de información de restablecimiento del POC no anidado"*.

[0208] De forma alternativa, la asociación del mensaje SEI se puede definir de la siguiente manera: *"Un mensaje SEI de información de restablecimiento del POC no anidado se asocia con la imagen picA que tiene el mismo valor nuh_layer_id que la unidad NAL de SEI, y tiene sucede, en el orden de descodificación, al mensaje SEI y precede a la primera imagen que tiene el mismo valor que nuh_layer_id como la unidad NAL de SEI y tiene los valores de poc_lsb_reset_flag o poc_msb_reset_flag como igual a 1."*

[0209] De forma alternativa, puede señalizarse un mensaje SEI vacío que indica la anulación de la información de restablecimiento del POC (*poc_reset_info_cancel()*) y la asociación del mensaje SEI se puede definir de la siguiente manera: *"Un mensaje SEI de información de restablecimiento del POC no anidado se asocia con la primera imagen picA que tiene el mismo valor de nuh_layer_id que la unidad NAL de SEI, que sucede al mensaje SEI en el orden de descodificación, y que está contenida en la unidad de acceso que contiene un mensaje SEI poc_reset_info_cancel(). El valor de nuh_layer_id de la unidad NAL de SEI que contiene el mensaje SEI poc_reset_info_cancel() será igual al nuh_layer_id de la imagen asociada"*.

[0210] La siguiente semántica se puede utilizar para *poc_reset_type_flag*, *poc_reset_value* y *poc_reset_id*: *"poc_reset_type_flag igual a 1 indica que POC MSB se restableció y que POC LSB no se restableció para la imagen de restablecimiento de POC asociada. poc_reset_type_flag igual a 0 especifica que tanto POC MSB como POC LSB se restablecieron para la imagen de restablecimiento de POC asociada; poc_reset_value indica el valor de POC de la imagen de restablecimiento de POC asociada antes de que se aplique el restablecimiento de POC (es decir, el valor de POC obtenido suponiendo que tanto poc_msb_reset_flag como poc_lsb_reset_flag sean iguales a 0), y poc_reset_id especifica un identificador de una imagen de restablecimiento de POC en la misma capa como la imagen asociada. Ninguna de las dos imágenes consecutivas de restablecimiento de POC de una capa particular en el flujo de bits tendrá el mismo valor de poc_reset_id"*.

[0211] Se debe tener en cuenta que, cuando se pierde la imagen de restablecimiento de POC asociada, este valor también se puede utilizar para obtener los valores POC de la imagen asociada y otras imágenes descodificadas de la misma capa en la DPB, de la siguiente manera. Cuando el valor de poc_reset_type_flag es igual a 0, el POC de la imagen asociada se puede obtener estableciendo prevPicOrderCntLsb igual a poc_reset_value % MaxPicOrderCntLsb, y prevPicOrderCntMsb igual a 0, y siguiendo el resto del proceso de descodificación para el recuento del orden de imágenes, y el valor de PicOrderCntVal de todas las imágenes en la DPB que pertenecen a la misma capa que la imagen asociada disminuyen en poc_reset_value - poc_reset_value % MaxPicOrderCntLsb. Cuando el valor de poc_reset_type_flag es igual a 1, el POC de la imagen asociada se puede obtener estableciendo prevPicOrderCntLsb y prevPicOrderCntMsb ambos iguales a 0, y siguiendo el resto del proceso de descodificación para el recuento del orden de imágenes y el valor de PicOrderCntVal de todas las imágenes en la DPB que pertenecen a la misma capa que la imagen asociada se disminuyen en poc_reset_value.

[0212] En algunos modos de realización, los elementos sintácticos similares a los descritos anteriormente en relación con los mensajes SEI se incluyen en la sintaxis de la cabecera de segmento del fragmento, y la frase "imagen actual" se usa en lugar de la frase "imagen asociada" en la semántica de ejemplo descrita anteriormente en relación con la semántica del mensaje SEI de información de restablecimiento del POC.

Otras Consideraciones

[0213] La información y las señales divulgadas en el presente documento pueden representarse usando cualquiera entre varias tecnologías y técnicas diferentes. Por ejemplo, los datos, las instrucciones, los comandos, la información, las señales, los bits, los símbolos y los chips que puedan haberse mencionado a lo largo de la descripción anterior pueden representarse mediante tensiones, corrientes, ondas electromagnéticas, campos o partículas magnéticos, campos o partículas ópticos o cualquier combinación de los mismos.

[0214] Los diversos bloques lógicos, módulos, circuitos y etapas de algoritmo ilustrativos, descritos en relación con los modos de realización divulgados en el presente documento pueden implementarse como hardware electrónico, software informático o combinaciones de ambos. Para ilustrar claramente esta intercambiabilidad de hardware y software, anteriormente se han descrito 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 la aplicación particular y de las restricciones de diseño impuestas en el sistema general. Los expertos en la materia 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.

[0215] Las técnicas descritas en el presente documento pueden implementarse en hardware, software, firmware o en cualquier combinación de lo anterior. Dichas técnicas pueden implementarse en cualquiera entre una variedad de dispositivos tales como ordenadores de propósito general, equipos manuales de dispositivos de comunicación inalámbrica o dispositivos de circuitos integrados que tienen múltiples usos, incluyendo su aplicación en equipos manuales de dispositivos de comunicación inalámbrica y otros dispositivos. Todas las características descritas como módulos o componentes pueden implementarse juntas en un dispositivo lógico integrado o por separado, como dispositivos lógicos discretos pero interoperables. Si se implementan en software, las técnicas pueden realizarse, al menos en parte, mediante un medio de almacenamiento de datos legible por ordenador que comprenda código de programa que incluye instrucciones que, cuando se ejecutan, realizan uno o más de los procedimientos descritos anteriormente. El medio de almacenamiento de datos legible por ordenador puede formar parte de un producto de programa informático, que puede incluir materiales de embalaje. El medio legible por ordenador puede comprender memoria o medios de almacenamiento de datos, tales como memoria de acceso aleatorio (RAM), tal como memoria de acceso aleatorio dinámica síncrona (SDRAM), memoria de solo lectura (ROM), memoria de acceso aleatorio no volátil (NVRAM), memoria de solo lectura programable y 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 a las que se puede acceder, leer y/o ejecutar por medio de un ordenador, tales como señales u ondas propagadas.

[0216] El código de programa puede ser ejecutado por un procesador, que puede incluir uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados específicos de la aplicación (ASIC), matrices de puertas programables en el terreno (FPGA) u otros circuitos lógicos equivalentes, integrados o discretos. Un procesador de este tipo puede estar configurado para realizar cualquiera de las técnicas descritas en esta divulgación. Un procesador de propósito general puede ser un microprocesador, pero, de forma alternativa, el procesador puede ser cualquier procesador, controlador, microcontrolador o máquina de estados convencional. Un procesador también 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 de dichas configuraciones. Por consiguiente, el término "procesador", como se usa en el presente documento, puede referirse a cualquier estructura anterior, cualquier combinación de la estructura anterior, o cualquier otra estructura

o aparato adecuados para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento puede proporcionarse dentro de módulos de 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 totalmente en uno o más circuitos o elementos lógicos.

5

[0217] Las técnicas de esta divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, incluidos un equipo manual inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). Diversos componentes, módulos o unidades se describen en esta divulgación para enfatizar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no requieren necesariamente su realización mediante diferentes unidades de hardware. En cambio, como se ha descrito anteriormente, diversas unidades se pueden combinar en una unidad de hardware de códec o proporcionarse por un grupo de unidades de hardware interoperativas, incluidos uno o más procesadores, como se ha descrito anteriormente, conjuntamente con software y/o firmware adecuados.

10

15

[0218] 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, comprendiendo el aparato:

5 una unidad de memoria configurada para almacenar información de vídeo asociada a una primera capa de vídeo que tiene una primera unidad de acceso; y

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

10 determinar (1005) si la primera unidad de acceso tiene al menos una imagen que es una imagen de punto de acceso aleatorio interno (IRAP) asociada con una primera indicación de que no se deben emitir imágenes principales omitidas de acceso aleatorio (RASL) de al menos una imagen; y

15 en respuesta a la determinación de que la primera unidad de acceso tiene al menos una imagen que es una imagen IRAP asociada con la primera indicación, proporcionar (1010) una segunda indicación, en un flujo de bits, para restablecer solo uno o más bits más significativos (MSB) de un recuento del orden de imágenes (POC) de cada imagen en la primera unidad de acceso que no es una imagen IRAP asociada con la primera indicación.

2. Un aparato configurado para descodificar información de vídeo, comprendiendo el aparato:

25 una unidad de memoria configurada para almacenar información de vídeo asociada a una primera capa de vídeo que tiene una primera unidad de acceso; y

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

30 determinar (1005) si la primera unidad de acceso tiene al menos una imagen que es una imagen de punto de acceso aleatorio interno (IRAP) asociada con una primera indicación de que no se deben emitir imágenes principales omitidas de acceso aleatorio (RASL) de al menos una imagen; y

35 en respuesta a la determinación de que la primera unidad de acceso tiene al menos una imagen que es una imagen IRAP asociada con la primera indicación, proporcionar (1010) una segunda indicación, en un flujo de bits, para restablecer solo uno o más bits más significativos (MSB) de un recuento del orden de imágenes (POC) de cada imagen en la primera unidad de acceso que no es una imagen IRAP asociada con la primera indicación.

3. El aparato según la reivindicación 1 o la reivindicación 2, en el que el procesador está configurado para señalar uno o más indicadores que indican que el uno o más MSB del POC de dicha imagen en la primera unidad de acceso que no es una imagen IRAP asociada con la primera indicación deben ser restablecidos.

4. El aparato según la reivindicación 1 o la reivindicación 2, en el que la primera capa de vídeo es una capa base, y en el que el procesador está configurado, además, para determinar si la primera unidad de acceso tiene una imagen en la primera capa de vídeo que es una imagen de actualización instantánea del descodificador (IDR), y en respuesta a la determinación de que la primera unidad de acceso tiene una imagen en la primera capa de vídeo que es una imagen IDR, señalar uno o más indicadores que indican que uno o más bits menos significativos (LSB) de cada imagen en la primera unidad de acceso la unidad que no es una imagen IDR deben restablecerse.

5. El aparato según la reivindicación 1 o la reivindicación 2, en el que el procesador está configurado además para señalar uno o más indicadores asociados con una o más imágenes que siguen una de cada una de dichas imágenes en el orden de descodificación y están en una misma capa de vídeo que la de cada una de dichas imágenes, indicando el uno o más indicadores que se debe restablecer un POC de cada una de las una o más imágenes.

6. El aparato según la reivindicación 5, en el que el procesador está configurado además para señalar uno o más indicadores indicando que se deben restablecer uno o más MSB del POC de cada una de las una o más imágenes.

7. El aparato según la reivindicación 1 o la reivindicación 2, en el que el procesador está configurado además para determinar si la primera unidad de acceso sigue inmediatamente un punto de empalme, y en el que el procesador está configurado para proporcionar la segunda indicación en el flujo de bits adicionalmente en respuesta a la determinación de que la primera unidad de acceso sigue inmediatamente a un punto de empalme.

8. El aparato según la reivindicación 1, en el que el aparato está configurado además para codificar la información de vídeo en el flujo de bits, o el
- 5 aparato según la reivindicación 2, en el que el aparato está configurado además para decodificar la información de vídeo en el flujo de bits.
9. El aparato según la reivindicación 1 o la reivindicación 2, en el que el aparato comprende un dispositivo seleccionado del grupo que consiste en ordenadores, portátiles, ordenadores portátiles, ordenadores de tableta, descodificadores, equipos telefónicos manuales, teléfonos inteligentes, paneles inteligentes,
- 10 televisores, cámaras, dispositivos de visualización., reproductores de medios digitales, consolas de videojuegos y ordenadores para automóvil.
10. Un procedimiento de codificación de información de vídeo, comprendiendo el procedimiento:
- 15 determinar (1005) si una primera unidad de acceso de una primera capa de vídeo tiene al menos una imagen que es una imagen de punto de acceso aleatorio interno (IRAP) asociada con una primera indicación de que no se deben emitir las imágenes principales omitidas de acceso aleatorio (RASL); y
- 20 en respuesta a la determinación de que la primera unidad de acceso tiene al menos una imagen que es una imagen IRAP asociada con la primera indicación, proporcionar (1010) una segunda indicación, en un flujo de bits, para restablecer solo uno o más bits más significativos (MSB) de un recuento del orden de imágenes (POC) de cada imagen en la primera unidad de acceso que no es una imagen IRAP asociada con la primera indicación.
- 25 11. El procedimiento según la reivindicación 10, en el que proporcionar la indicación en el flujo de bits comprende señalar uno o más indicadores que indican que el uno o más MSB del POC de cada una de dichas imágenes en la primera unidad de acceso que no es una imagen IRAP asociada con la primera indicación deben ser restablecidos.
- 30 12. El procedimiento según la reivindicación 10, en el que la primera capa de vídeo es una capa base, y en el que el procedimiento comprende además determinar si la primera unidad de acceso tiene una imagen en la primera capa de vídeo que es una imagen de actualización instantánea del descodificador (IDR), y en respuesta a determinar que la primera unidad de acceso tiene una imagen en la primera capa de vídeo que es una imagen IDR, señalizando uno o más indicadores que indican que uno o más bits menos significativos (LSB) de cada imagen en la primera unidad de acceso que no es una imagen IDR deben ser restablecidos.
- 35 13. El procedimiento según la reivindicación 10, que comprende además señalar uno o más indicadores asociados con una o más imágenes que siguen a una de cada una de dichas imágenes en el orden de decodificación y están en la misma capa de vídeo que la de cada una de dichas imágenes, indicando el uno o más indicadores que se debe restablecer un POC de cada una de las una o más imágenes,
- 40 en el que la señalización de uno o más indicadores comprende preferentemente indicar que uno o más MSB del POC de cada una de las una o más imágenes deben restablecerse.
- 45 14. El procedimiento según la reivindicación 10, que comprende además determinar si la primera unidad de acceso sigue inmediatamente a un punto de empalme, en el que proporcionar la segunda indicación comprende proporcionar la segunda indicación en el flujo de bits adicionalmente en respuesta a determinar que la primera unidad de acceso sigue inmediatamente a un punto de empalme.
- 50 15. Un medio no transitorio legible por ordenador que comprende código que, cuando se ejecuta, hace que un aparato realice el procedimiento de cualquiera de las reivindicaciones 10 a 14.

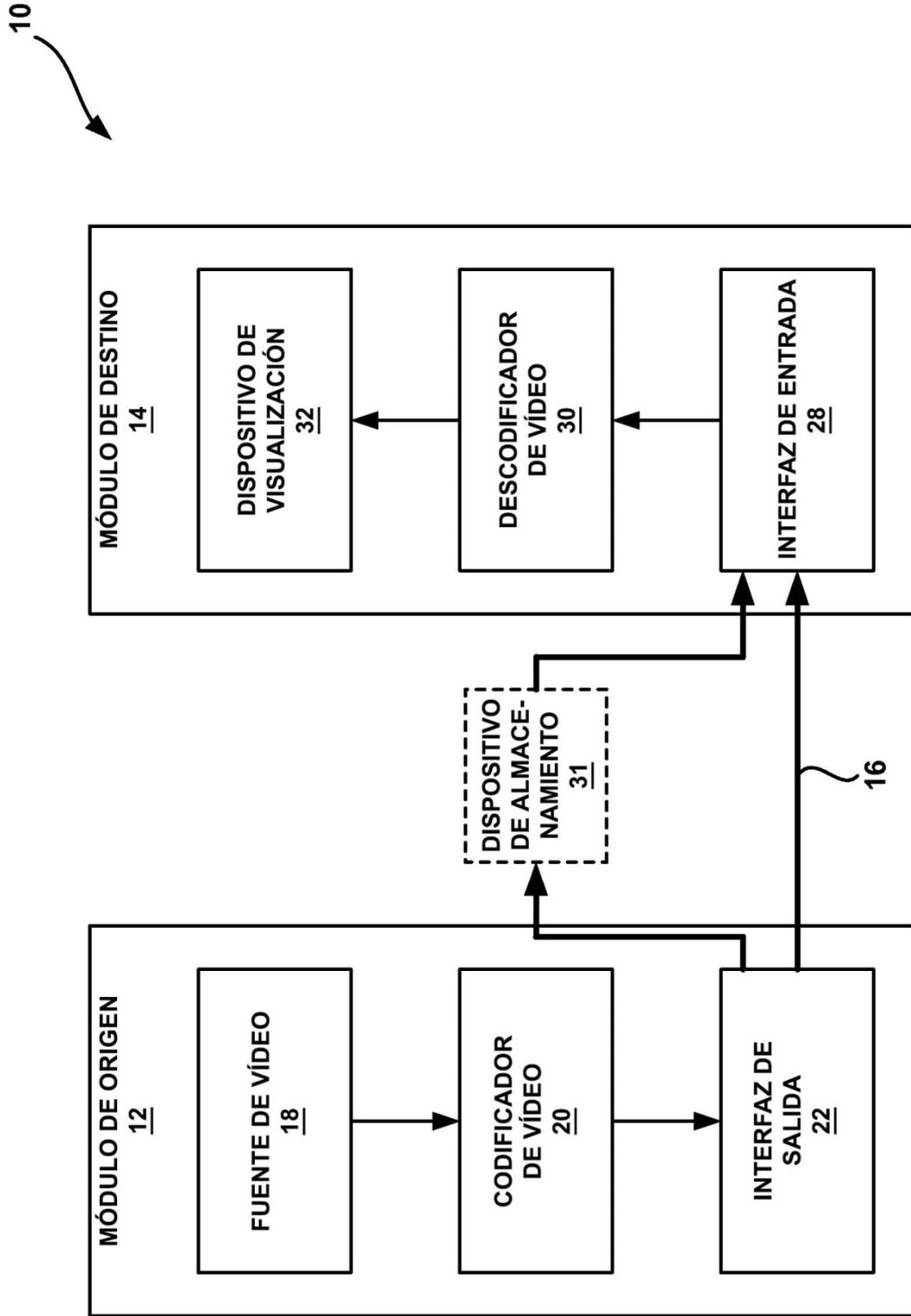


FIG. 1A

10'

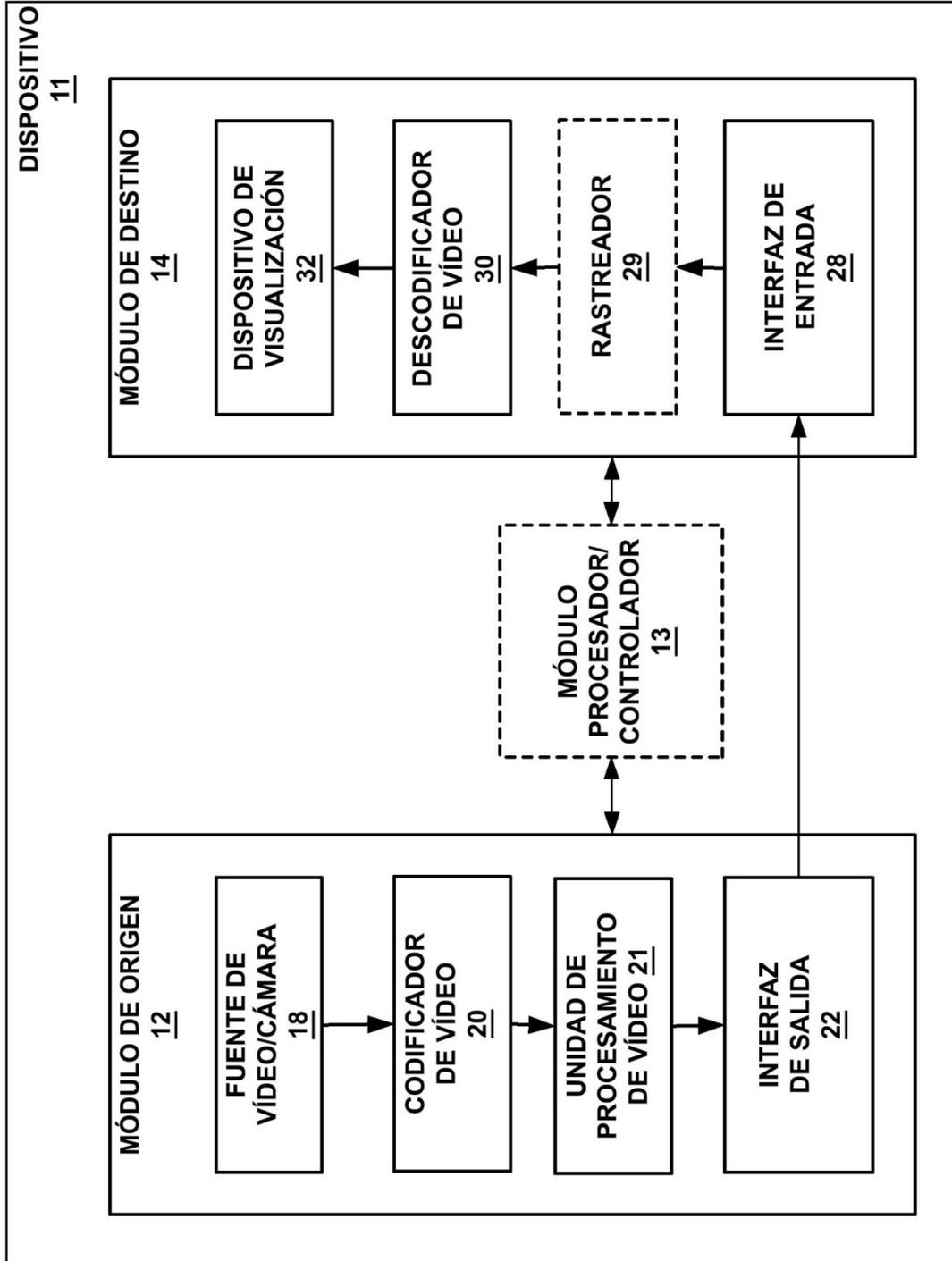


FIG. 1B

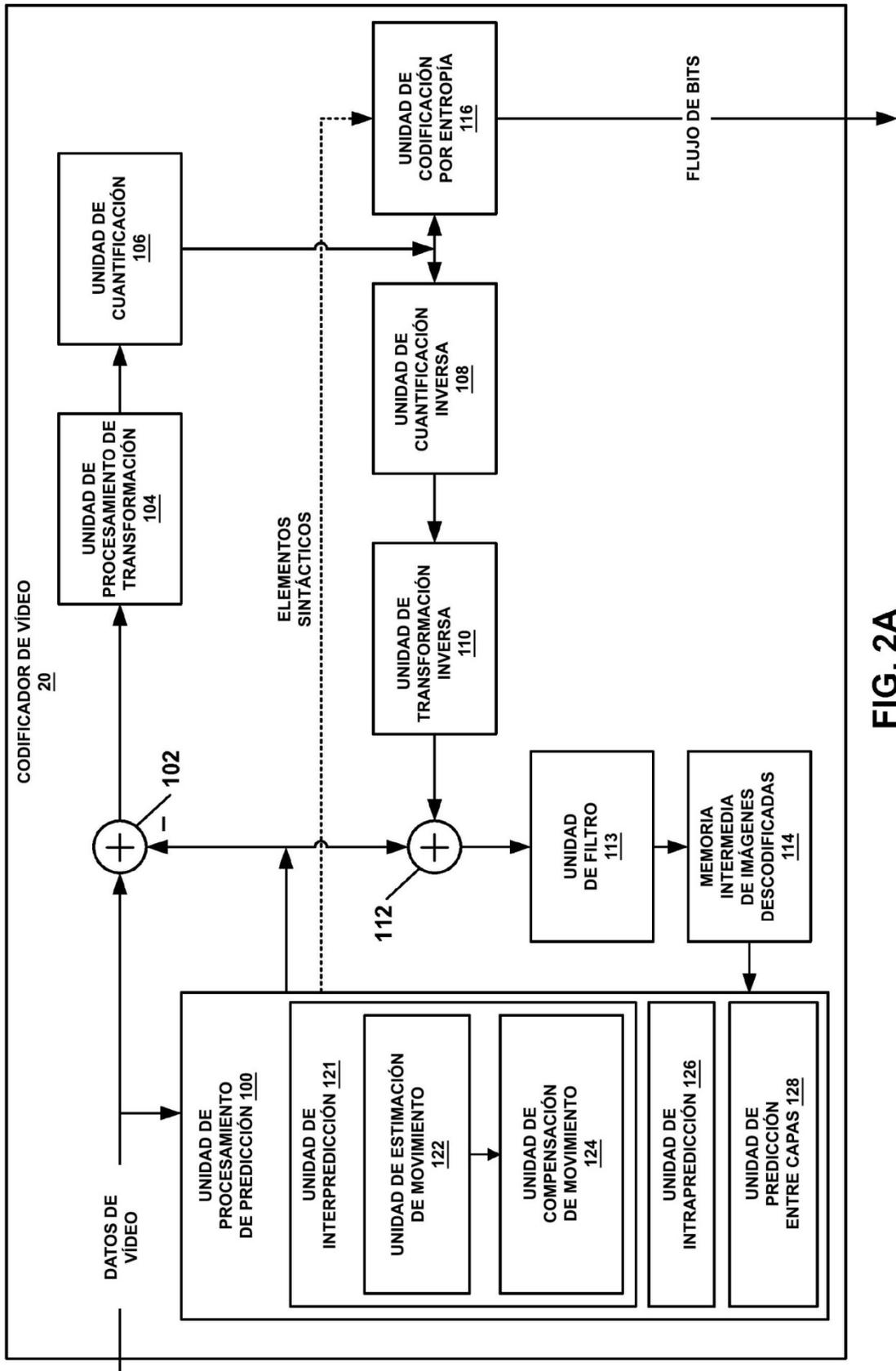


FIG. 2A

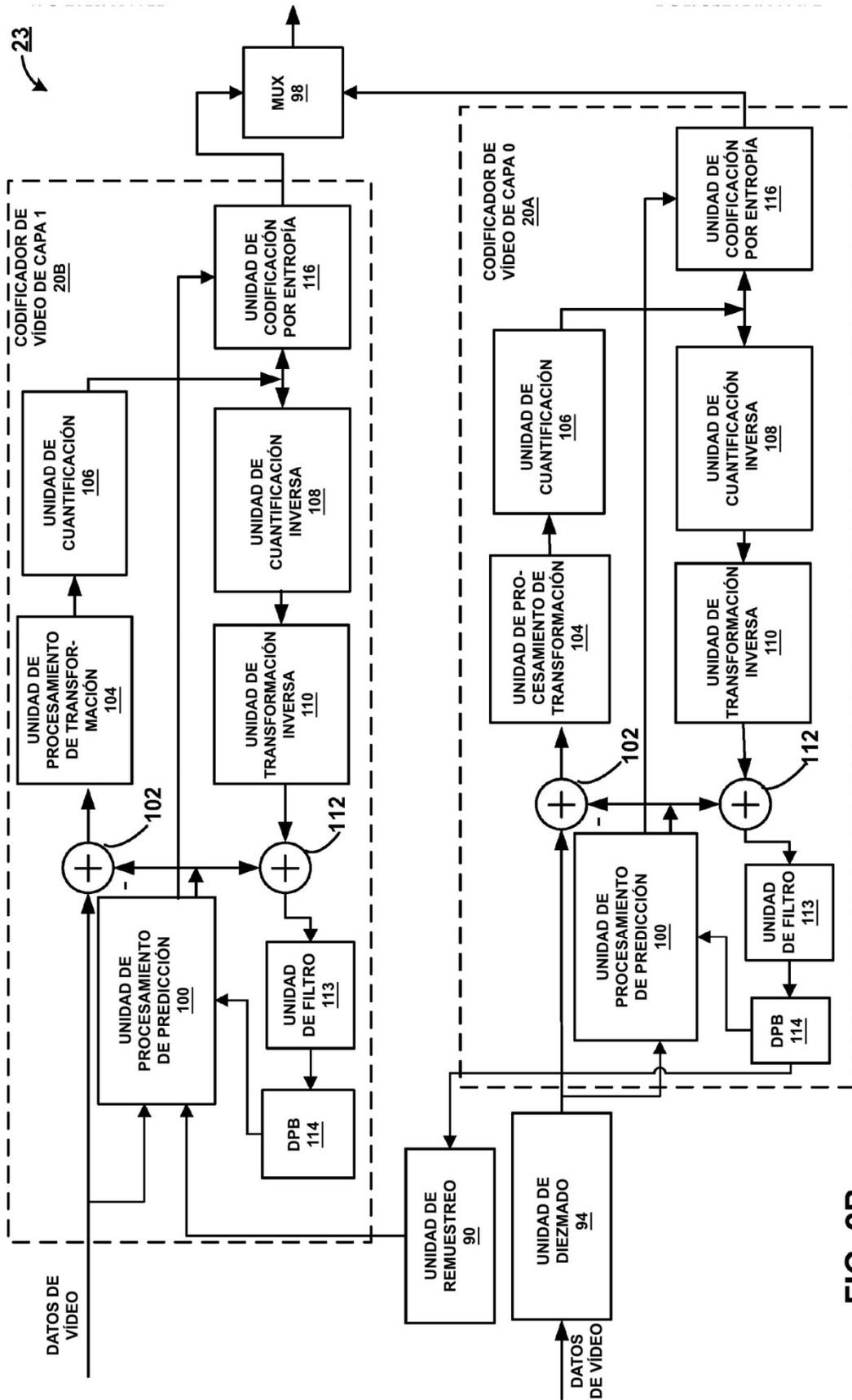


FIG. 2B

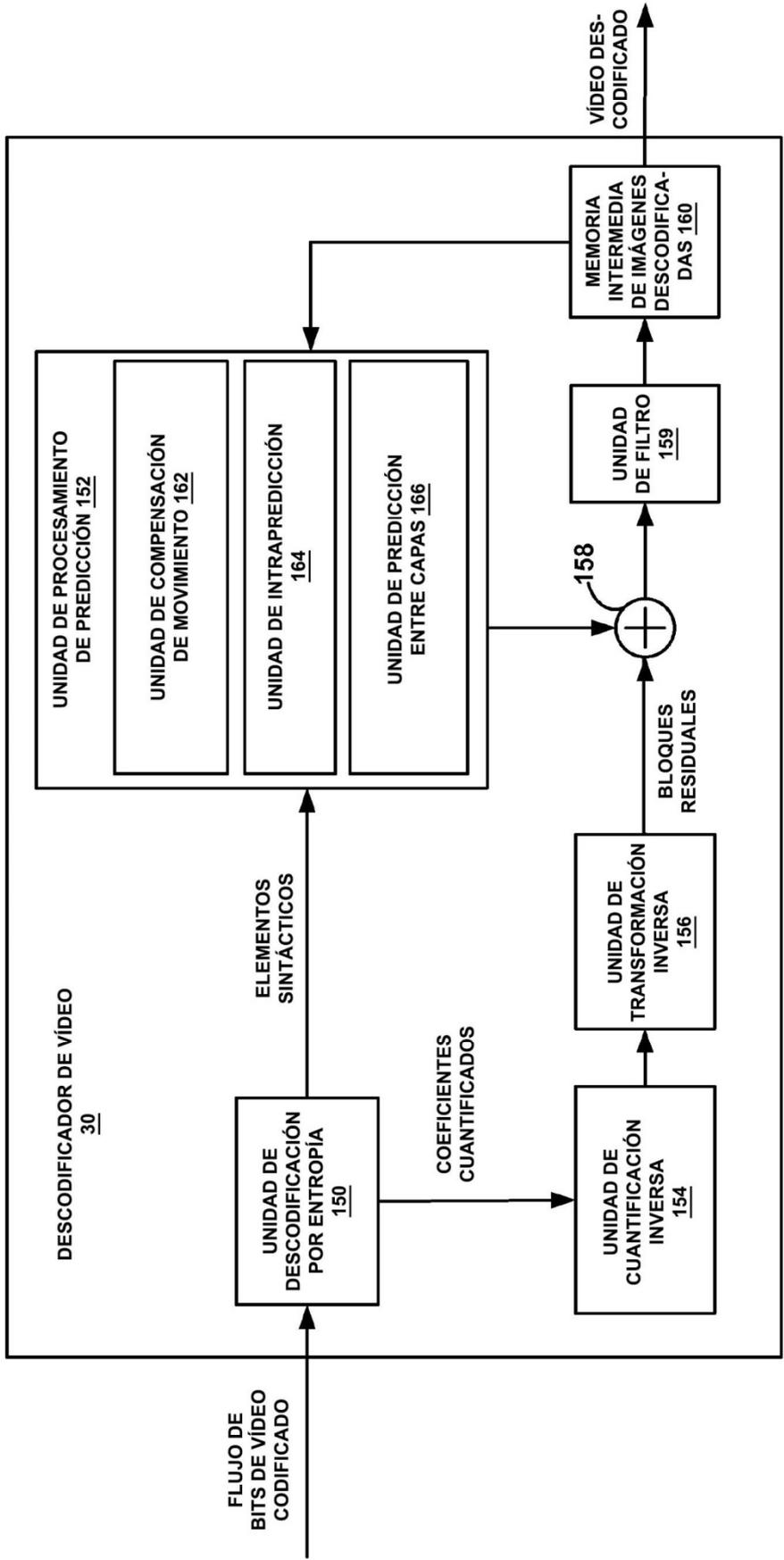


FIG. 3A

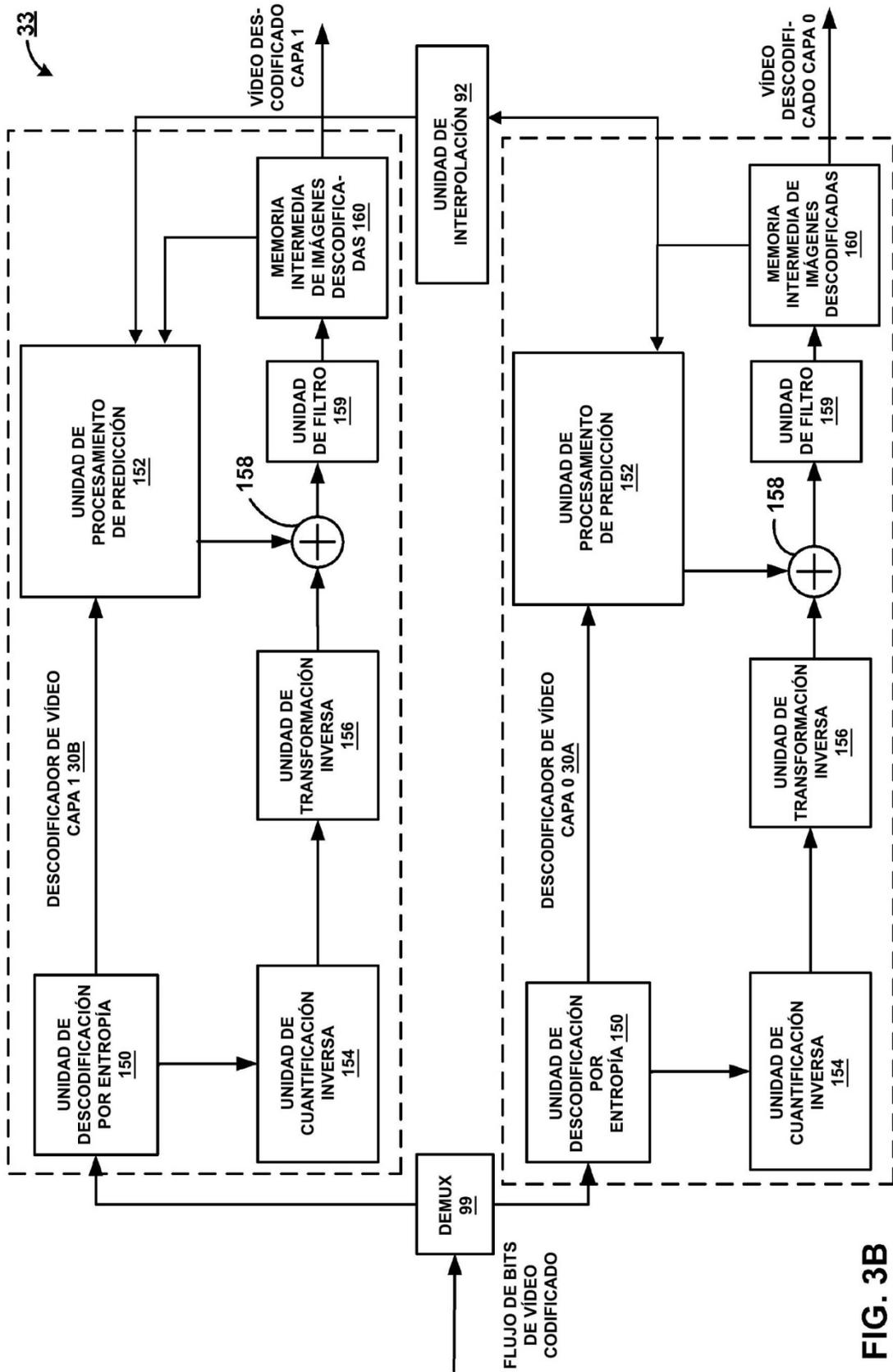


FIG. 3B

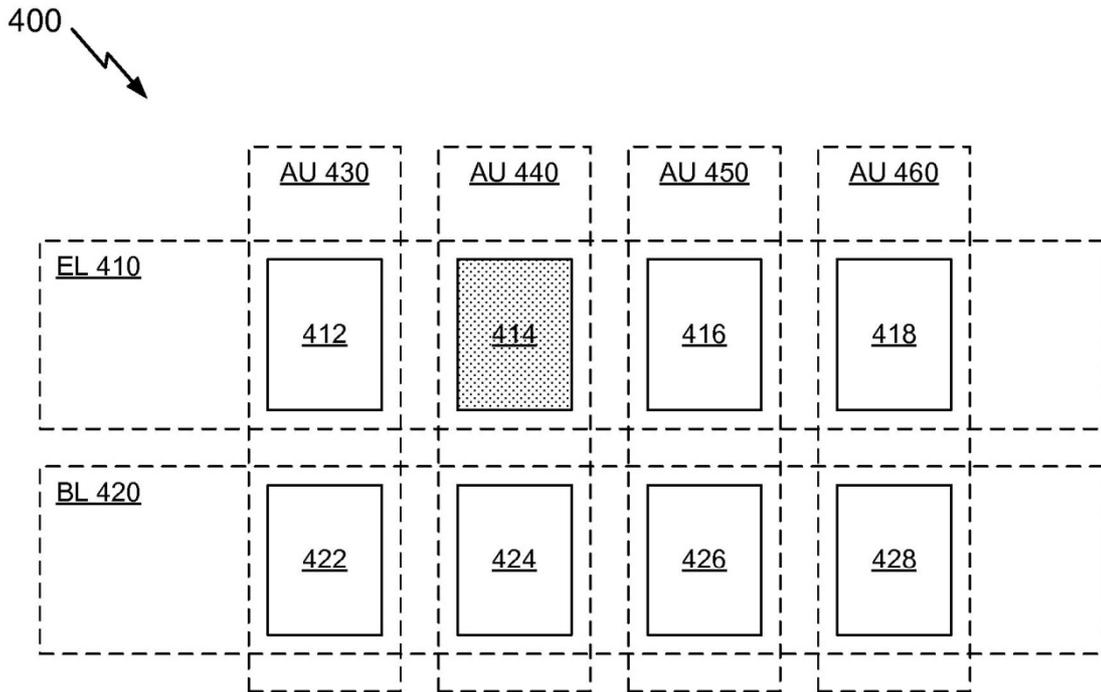
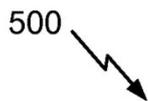


FIG. 4



	AU		430	440	450	460
SIN RESTABLE- CIMIENTO	LSB	EL	112	113	114	115
		BL	112	113	114	115
	MSB	EL	1	0	0	0
		BL	1	1	1	1
CON RESTABLE- CIMIENTO	LSB	EL	112	113	114	115
		BL	112	113	114	115
	MSB	EL	1	0	0	0
		BL	1	0	0	0

FIG. 5

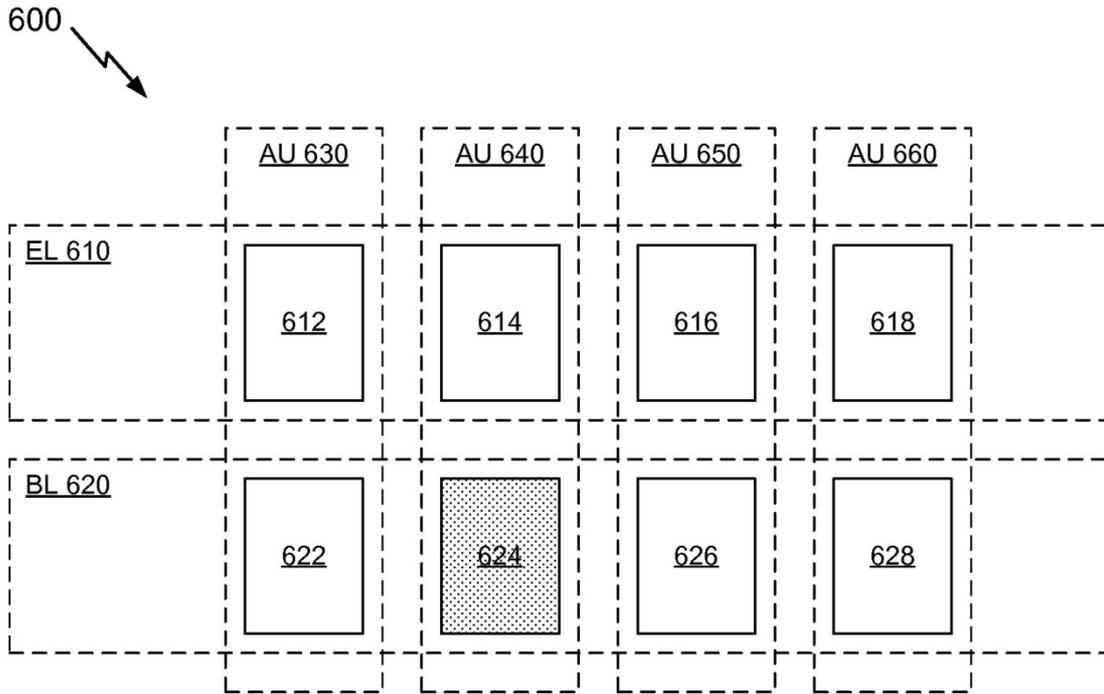
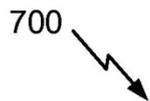


FIG. 6



	AU		630	640	650	660
SIN RESTABLECIMIENTO	LSB	EL	112	113	114	115
		BL	112	0	1	2
	MSB	EL	1	1	1	1
		BL	1	0	0	0
CON RESTABLECIMIENTO	LSB	EL	112	0	1	2
		BL	112	0	1	2
	MSB	EL	1	0	0	0
		BL	1	0	0	0

FIG. 7

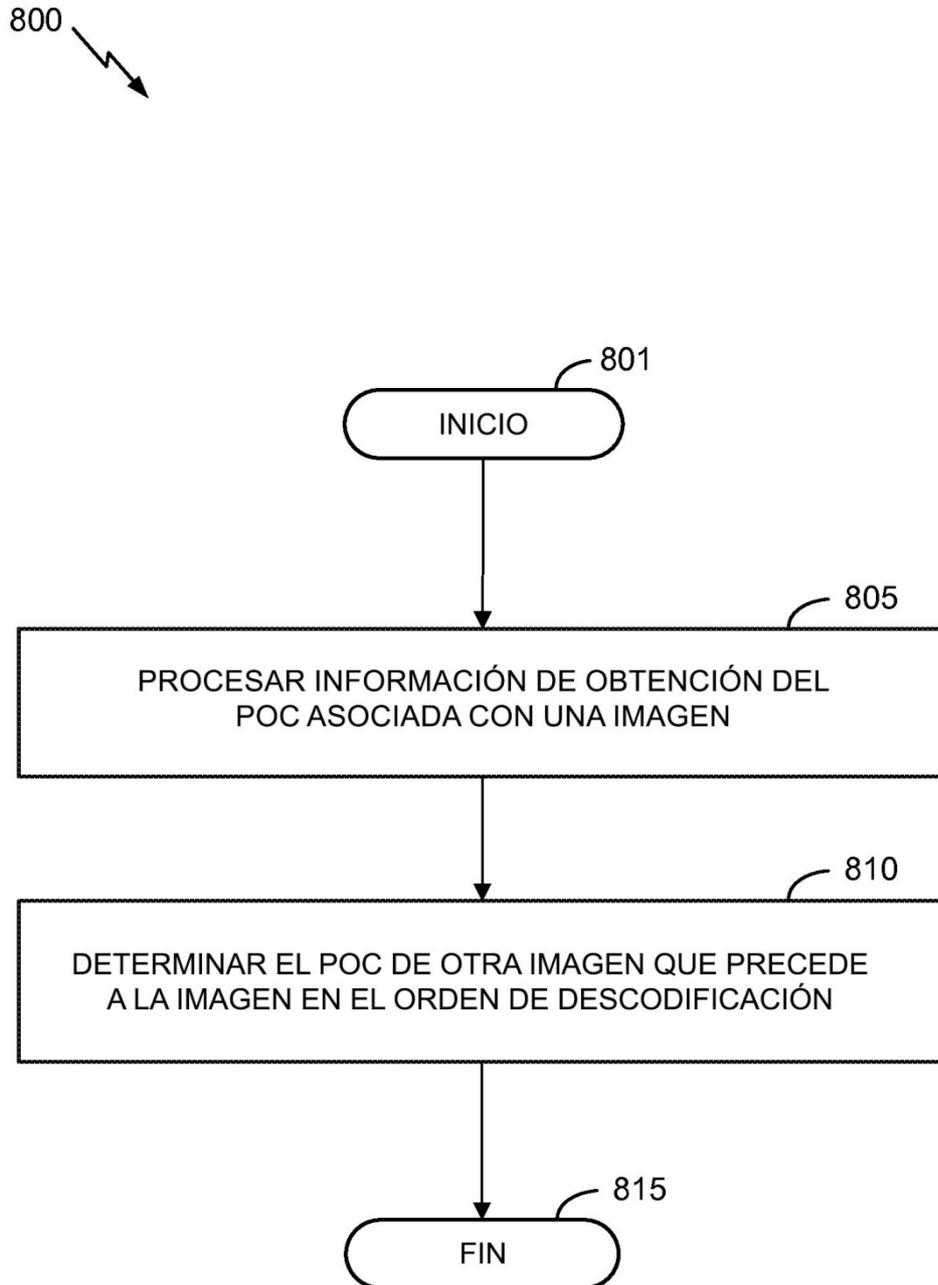


FIG. 8

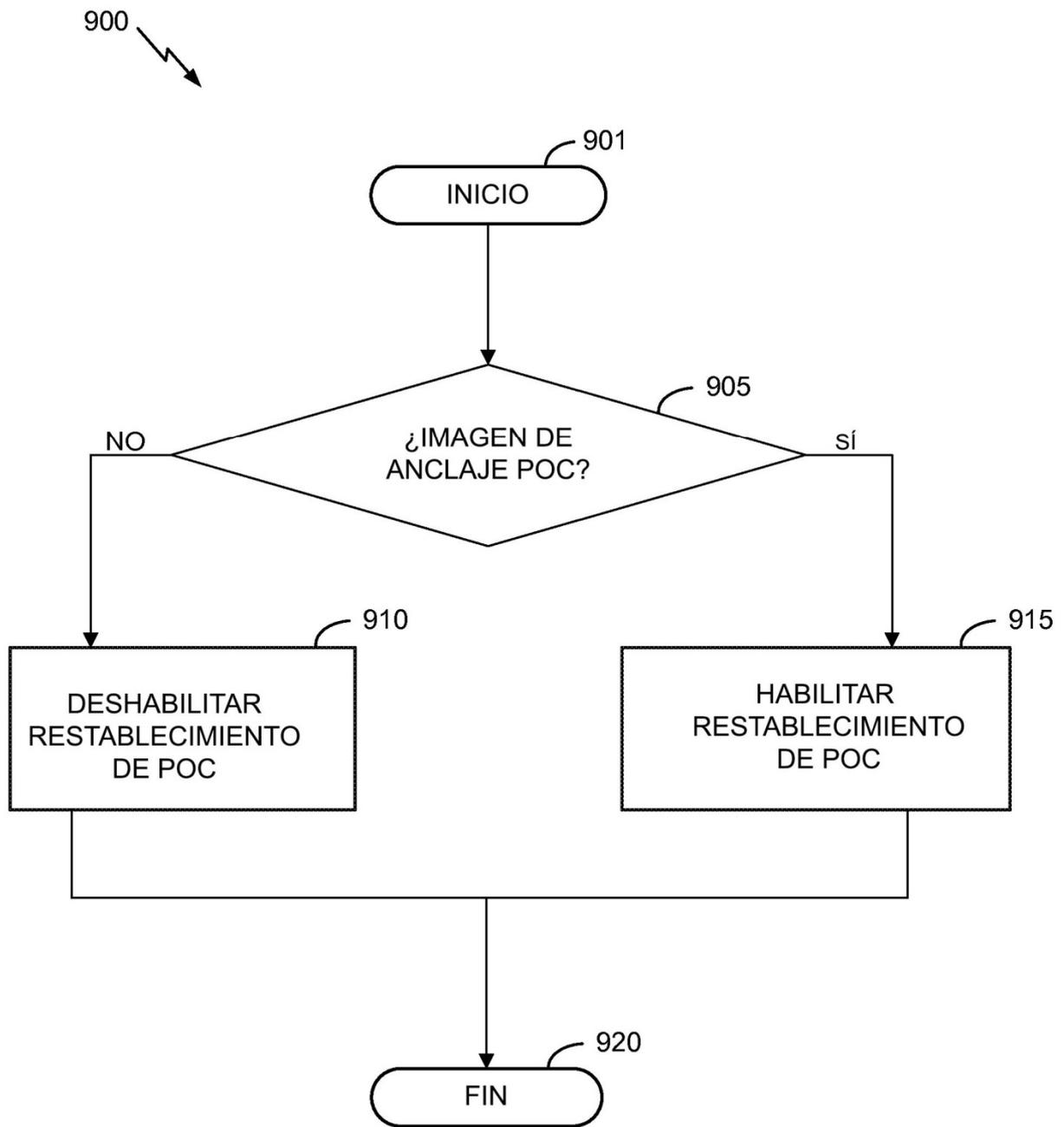


FIG. 9

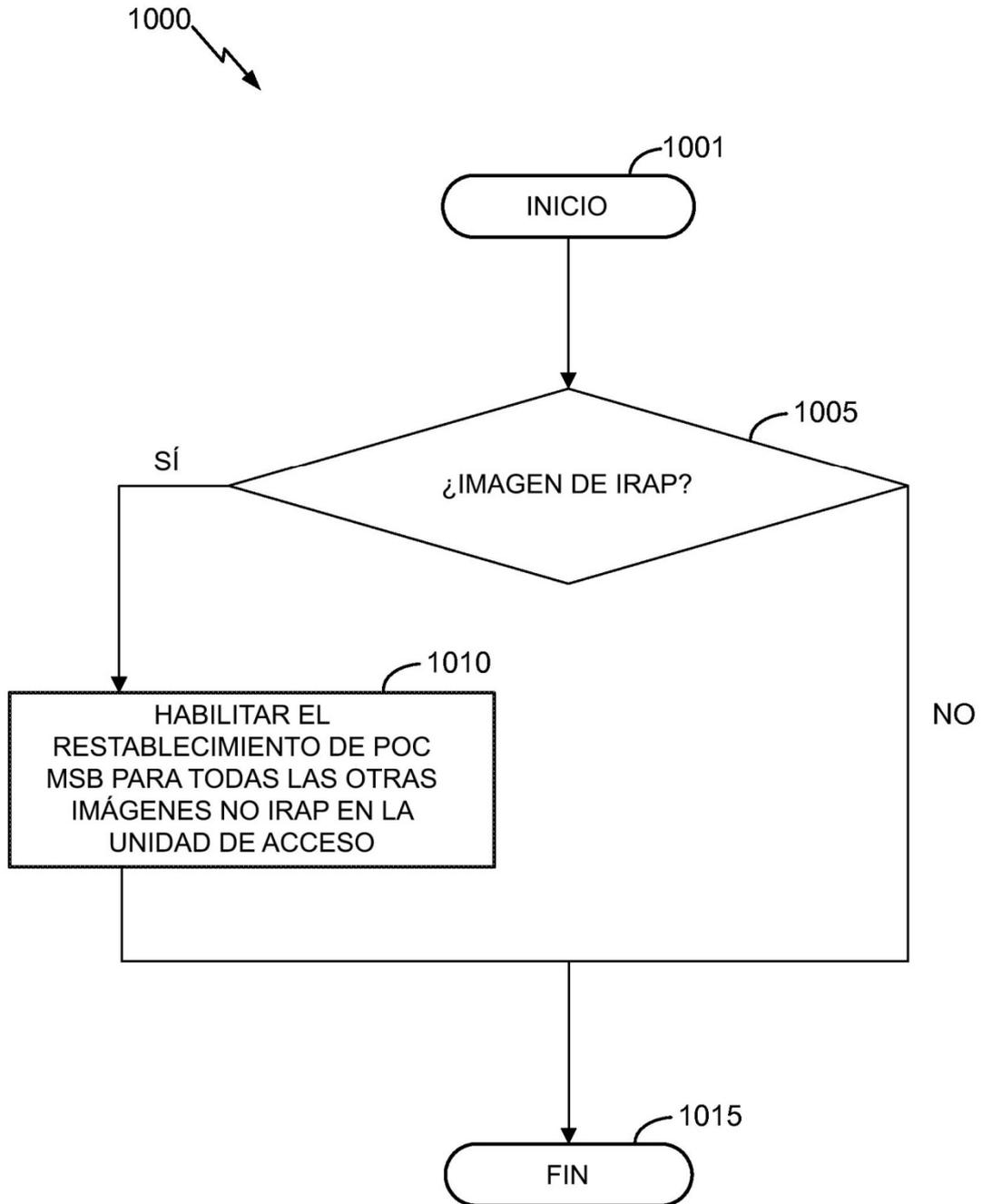


FIG. 10

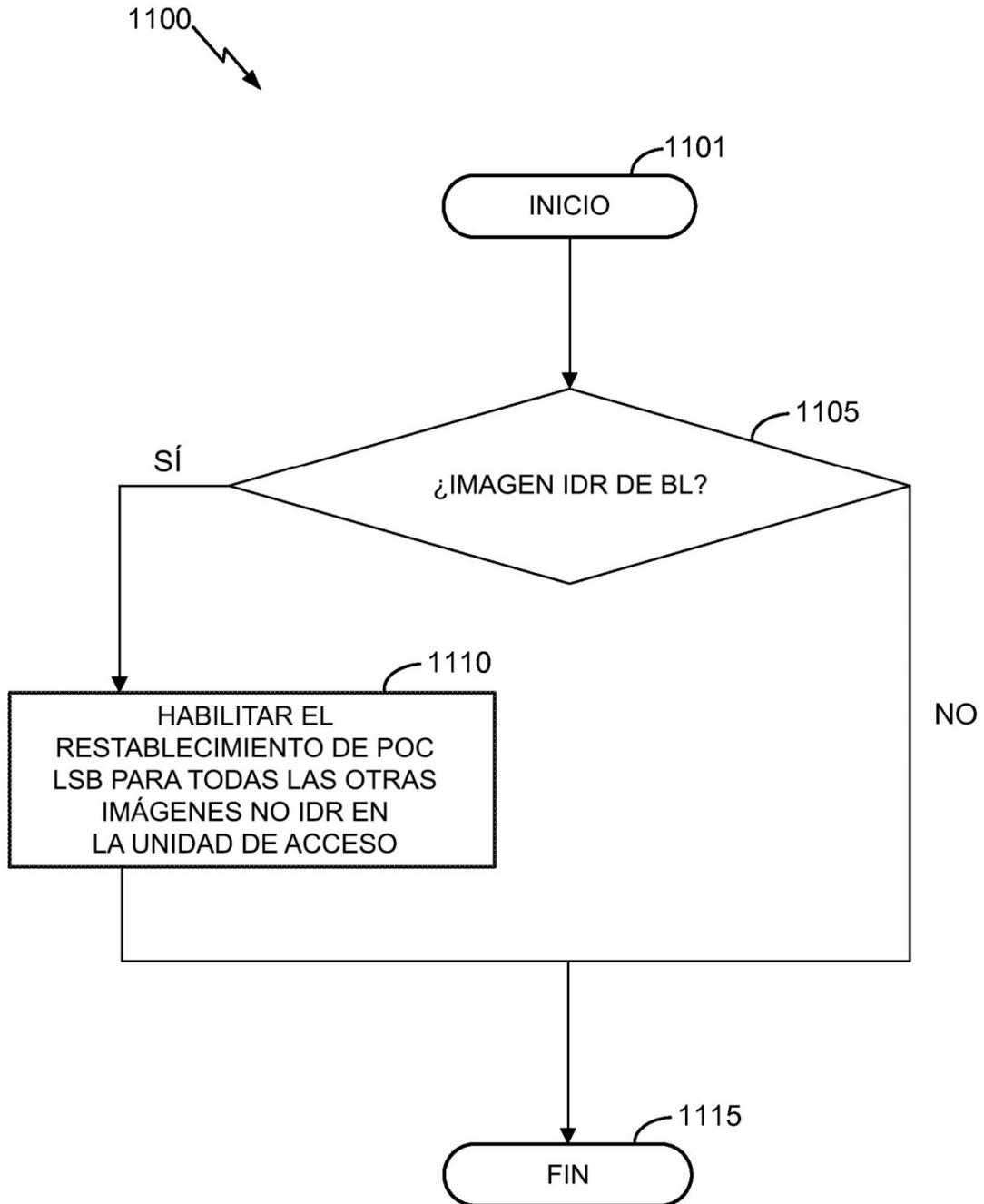


FIG. 11

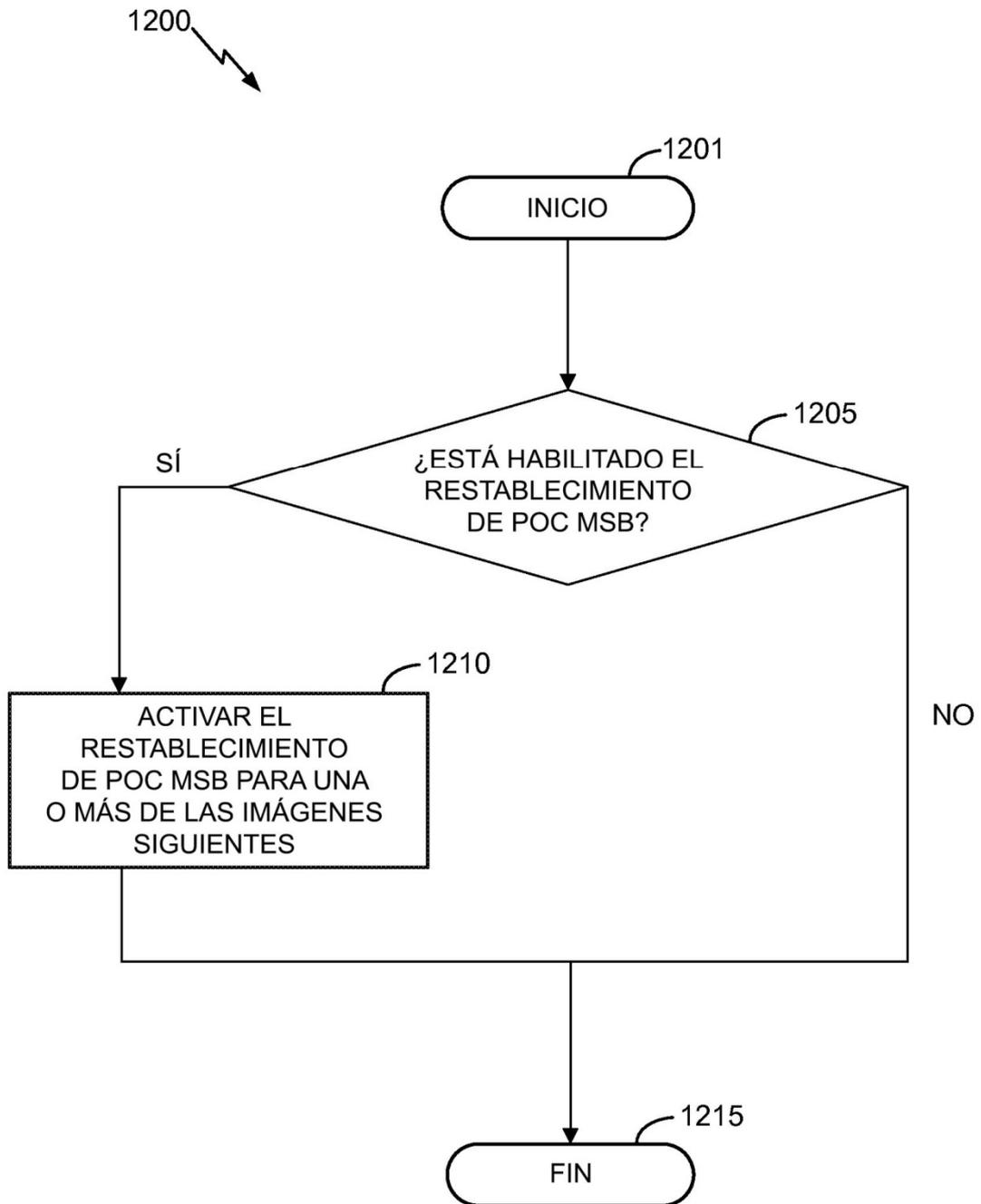


FIG. 12