

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 773 276**

51 Int. Cl.:

H04N 19/70 (2014.01)

H04N 19/30 (2014.01)

H04N 19/46 (2014.01)

H04N 19/597 (2014.01)

H04N 19/68 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **07.04.2014 PCT/US2014/033172**

87 Fecha y número de publicación internacional: **16.10.2014 WO14168872**

96 Fecha de presentación y número de la solicitud europea: **07.04.2014 E 14724258 (0)**

97 Fecha y número de publicación de la concesión europea: **13.11.2019 EP 2984844**

54 Título: **Alineación POC de capa cruzada para flujos de bits multicapa que pueden incluir imágenes de IRAP no alineadas**

30 Prioridad:

08.04.2013 US 201361809855 P

15.07.2013 US 201361846532 P

04.04.2014 US 201414245115

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

10.07.2020

73 Titular/es:

QUALCOMM INCORPORATED (100.0%)

5775 Morehouse Drive

San Diego, CA 92121-1714, US

72 Inventor/es:

CHEN, YING;

WANG, YE-KUI y

RAMASUBRAMONIAN, ADARSH KRISHNAN

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 773 276 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

Alineación POC de capa cruzada para flujos de bits multicapa que pueden incluir imágenes de IRAP no alineadas

5 **CAMPO TÉCNICO**

[0001] Esta divulgación se refiere a la codificación de vídeo.

10 **ANTECEDENTES**

10 [0002] Las capacidades del vídeo digital pueden incorporarse en una amplia gama de dispositivos, incluyendo
 15 televisores digitales, sistemas de difusión directa digital, sistemas de difusión inalámbrica, asistentes digitales
 personales (PDA), ordenadores portátiles o de escritorio, tabletas electrónicas, lectores de libros electrónicos, cámaras
 digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de videojuegos, consolas
 20 de videojuegos, teléfonos celulares o de radio por satélite, los denominados "teléfonos inteligentes", dispositivos de
 videoconferencia, dispositivos de transmisión continua de vídeo y similares. Los dispositivos de vídeo digital
 implementan técnicas de codificación de vídeo, tales como las descritas en las normas definidas por MPEG-2, MPEG-
 4, ITU-T H.263, ITU-T H.264/MPEG-4, Parte 10, Codificación Avanzada de Vídeo (AVC), la norma de Codificación de
 Vídeo de Alta Eficacia (HEVC), actualmente en desarrollo, y las ampliaciones de tales normas. Los dispositivos de
 vídeo pueden transmitir, recibir, codificar, descodificar y/o almacenar información de vídeo digital más eficazmente,
 implementando dichas técnicas de codificación de vídeo.

25 [0003] Las técnicas de codificación de vídeo incluyen la predicción espacial (intraimagen) y/o la predicción temporal
 (interimagen) para reducir o eliminar la redundancia intrínseca en las secuencias de vídeo. Para la codificación de
 vídeo basada en bloques, un fragmento de vídeo (por ejemplo, una trama de vídeo o una porción de una trama de
 vídeo) puede dividirse en bloques de vídeo, que también pueden denominarse bloques de árbol, unidades de
 30 codificación (CU) y/o nodos de codificación. Los bloques de vídeo de un sector intracodificado (I) de una imagen se
 codifican usando predicción espacial con respecto a unas muestras de referencia de bloques vecinos de la misma
 imagen. Los bloques de vídeo de un sector intercodificado (P o B) de una imagen pueden usar predicción espacial
 con respecto a unas muestras de referencia de bloques vecinos de la misma imagen, o predicción temporal con
 respecto a unas muestras de referencia de otras imágenes de referencia. Las imágenes pueden denominarse tramas
 y las imágenes de referencia pueden denominarse tramas de referencia.

35 [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.
 40 Para una mayor compresión, los datos residuales pueden transformarse desde el dominio del píxel a un dominio de
 transformada, dando como resultado coeficientes de transformada residuales, que a continuación se pueden
 cuantificar. Los coeficientes de transformada cuantificados, dispuestos inicialmente en una formación bidimensional,
 pueden explorarse con el fin de generar un vector unidimensional de coeficientes de transformada, y puede aplicarse
 codificación por entropía para lograr aún más compresión.

45 [0005] En la publicación de Byeongdo Choi y col. titulada "3D-HEVC HLS: On Picture order counts" (100ª reunión
 de MPEG, Ginebra 2012, documento MPEG n.º M24874, 27 de abril de 2012), se divulga una restricción propuesta
 para el uso de imágenes IDR y CRA y el esquema de sincronización POC para el POC de multivistas. El mismo valor
 de POC debe mantenerse en todas las vistas de una unidad de acceso. Además, también se afirma que el tipo de
 50 imagen IDR debe usarse en una vista no base cuando el tipo de imagen de la vista base es IDR. El uso del tipo de
 imagen IDR en vistas no base también está prohibido cuando el tipo de imagen de la vista base no es IDR, para la
 sincronización POC entre vistas.

SUMARIO

55 [0006] En general, esta divulgación describe técnicas para codificar un elemento sintáctico para una imagen que
 indica si se debe restablecer un valor de contador del orden de imágenes (POC) para la imagen. En particular, un
 codificador de vídeo (como un codificador de vídeo o un descodificador de vídeo) puede codificar un valor que indica
 que un valor de POC para una imagen con punto de acceso no intraaleatorio (IRAP) se debe restablecer cuando la
 60 imagen no IRAP es incluida en una unidad de acceso con una imagen IRAP, por ejemplo, en diferentes capas de
 codificación de vídeo. De esta manera, las técnicas de esta divulgación pueden admitir imágenes IRAP sin alineación
 a través de diferentes capas de codificación de vídeo, al mismo tiempo que aseguran que los valores de POC estén
 alineados entre las imágenes de capas de codificación de vídeo diferentes.

65 [0007] De acuerdo con un primer aspecto, se proporciona un procedimiento para descodificar datos de vídeo, el
 procedimiento comprende: recibir una unidad de acceso que incluye un conjunto de primeras imágenes en las
 primeras capas de codificación de vídeo respectivas y una segunda imagen en una segunda capa de codificación de

vídeo diferente de las primeras capas de codificación de vídeo, en el que el conjunto de primeras imágenes incluye todas las imágenes en la unidad de acceso que no son imágenes de actualización instantánea del descodificador, IDR, y la segunda imagen es una imagen IDR; y para cada primera imagen del conjunto de primeras imágenes: i) recibir un elemento sintáctico indicativo de si al menos una porción de un valor del contador del orden de imágenes, POC, de la primera imagen se debe restablecer en un valor de cero, ii) cuando el valor para el elemento sintáctico indica que la porción del valor de POC se debe restablecer en el valor de cero, restablecer al menos la porción del valor de POC de modo que la porción del valor de POC sea igual a cero, en el que, antes de restablecer el valor de POC, el valor de POC para la primera imagen es diferente de un segundo valor de POC para la segunda imagen, y iii) descodificar datos de vídeo utilizando el valor de restablecimiento de POC, en el que un POC es una variable asociada con una imagen que identifica de forma única la imagen asociada entre imágenes en una secuencia de vídeo codificada, y en el que una unidad de acceso incluye todas las imágenes codificadas asociadas con el mismo tiempo de salida.

[0008] De acuerdo con un aspecto adicional, se proporciona un procedimiento para codificar datos de vídeo, el procedimiento que comprende: formar una unidad de acceso que incluye un conjunto de primeras imágenes en las primeras capas de codificación de vídeo respectivas y una segunda imagen en una segunda capa de codificación de vídeo, en el que la segunda capa de codificación de vídeo es diferente de las primeras capas de codificación de vídeo, en el que el conjunto de primeras imágenes incluye todas las imágenes en la unidad de acceso que no son imágenes de actualización instantánea del descodificador, IDR, y la segunda imagen es una imagen IDR; y para cada primera imagen del conjunto de primeras imágenes: i) determinar si se restablece al menos una porción de un valor de contador del orden de imágenes, POC, de la primera imagen en un valor de cero, ii) en respuesta a la determinación de restablecer la al menos porción del valor de POC, restablecer al menos la porción del valor de POC, de modo que la porción del valor de POC sea igual a cero, en el que, antes de restablecer el valor de POC, el valor de POC para la primera imagen es diferente de un segundo valor de POC para la segunda imagen, iii) codificar un valor para un elemento sintáctico que indica que al menos la porción del valor de POC se debe restablecer en el valor de cero, y iv) codificar datos de vídeo utilizando el valor de restablecimiento de POC, en el que un POC es una variable asociada con una imagen que identifica de forma única la imagen asociada entre imágenes en una secuencia de vídeo codificada, y en el que una unidad de acceso incluye todas las imágenes codificadas asociadas con el mismo tiempo de salida.

[0009] De acuerdo con un aspecto adicional, se proporciona un dispositivo para descodificar datos de vídeo, el dispositivo que comprende uno o más procesadores configurados para: recibir una unidad de acceso que incluye un conjunto de primeras imágenes en las primeras capas de codificación de vídeo respectivas y una segunda imagen en una segunda capa de codificación de vídeo diferente de la primera capa de codificación de vídeo, en el que el conjunto de primeras imágenes incluye todas las imágenes en la unidad de acceso que no son imágenes de actualización instantánea del descodificador, IDR, y la segunda imagen es una imagen IDR; y para cada primera imagen del conjunto de primeras imágenes: i) recibir un elemento sintáctico indicativo de si al menos una porción del valor de contador del orden de imágenes, POC, de la primera imagen se debe restablecer en un valor de cero; ii) cuando el valor para el elemento sintáctico indica que la porción del valor de POC se debe restablecer en el valor de cero, restablecer al menos la porción del valor de POC de modo que la porción del valor de POC sea igual a cero, en el que, antes de restablecer el valor de POC, el valor de POC para la primera imagen es diferente de un segundo valor de POC para la segunda imagen, y iii) descodificar datos de vídeo utilizando el valor de restablecimiento de POC, en el que un POC es una variable asociada con una imagen que identifica de forma exclusiva la imagen asociada entre imágenes en una secuencia de vídeo codificada, y en la que una unidad de acceso incluye todas las imágenes codificadas asociadas con el mismo tiempo de salida.

[0010] De acuerdo con un aspecto adicional, se proporciona un dispositivo para codificar datos de vídeo, el dispositivo que comprende uno o más procesadores configurados para: formar una unidad de acceso que incluye un conjunto de primeras imágenes en las primeras capas de codificación de vídeo respectivas y una segunda imagen en una segunda capa de codificación de vídeo, en el que la segunda capa de codificación de vídeo es diferente de las primeras capas de codificación de vídeo, en el que el conjunto de primeras imágenes incluye todas las imágenes en la unidad de acceso que no son imágenes de una actualización instantánea del descodificador, IDR, y la segunda imagen es una imagen IDR; y para cada primera imagen del conjunto de primeras imágenes: i) determinar si se restablece al menos una porción de un valor de contador del orden de imágenes, POC, de la primera imagen en un valor de cero, ii) en respuesta a la determinación de restablecer la al menos porción del valor de POC, restablecer al menos la porción del valor de POC, de modo que la porción del valor de POC sea igual a cero, en el que, antes de restablecer el valor de POC, el valor de POC para la primera imagen sea diferente de un segundo valor de POC para la segunda imagen, iii) codificar un valor para un elemento sintáctico que indica que al menos la porción del valor de POC se debe restablecer en el valor de cero, y iv) codificar datos de vídeo utilizando el valor de restablecimiento de POC, en el que un POC es una variable asociada con una imagen que identifica de forma única la imagen asociada entre imágenes en una secuencia de vídeo codificada, y en el que una unidad de acceso incluye todas las imágenes codificadas asociadas con el mismo tiempo de salida.

[0011] En otro ejemplo, un medio de almacenamiento legible por ordenador ha almacenado en el mismo instrucciones que, cuando se ejecutan, hacen que un procesador programable lleve a cabo uno de los procedimientos descritos anteriormente.

[0012] Los detalles de uno o más ejemplos se exponen en los dibujos adjuntos y en la descripción siguiente. Otras características, objetos y ventajas resultarán evidentes a partir de la descripción y de los dibujos, y a partir de las reivindicaciones.

5

BREVE DESCRIPCIÓN DE LOS DIBUJOS

[0013]

10 La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación y decodificación de vídeo de ejemplo de acuerdo con uno o más ejemplos descritos en esta divulgación.

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

15

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

20

La FIG. 4 es un diagrama conceptual que ilustra una secuencia de imágenes de vídeo codificadas.

25

La FIG. 5 es un diagrama de flujo que ilustra un procedimiento de codificación de datos de vídeo de ejemplo de acuerdo con las técnicas de esta divulgación.

30

La FIG. 6 es un diagrama de flujo que ilustra un procedimiento de decodificación de datos de vídeo de ejemplo de acuerdo con las técnicas de esta divulgación.

35

DESCRIPCIÓN DETALLADA

30 **[0014]** Esta divulgación describe técnicas para admitir imágenes con punto de acceso intraaleatorio no alineadas (IRAP) que pueden iniciar una nueva secuencia de vídeo codificada de manera que se garantice que los mismos valores de contador del orden de imágenes (POC) estén alineados en capa cruzada. Las técnicas pueden aplicarse en la codificación multicapa. Algunos procedimientos divulgados también se pueden aplicar a la codificación de vídeo de una sola capa.

35 **[0015]** En particular, las imágenes IRAP no alineadas se producen en situaciones en las que una unidad de acceso incluye tanto una imagen IRAP como una imagen no IRAP en diferentes capas de codificación de vídeo. Las imágenes en una secuencia de vídeo codificada (CVS) tienen valores de POC que identifican de forma única las imágenes respectivas en esa CVS. Los valores de POC de las imágenes en general indican órdenes de salida de las imágenes en relación con otras imágenes en la misma CVS. Las imágenes IRAP tienen valores de POC de 0. Por lo tanto, las imágenes IRAP no alineadas conllevan unidades de acceso que incluyen imágenes que tienen diferentes valores de POC.

40

45 **[0016]** Sin embargo, los valores de POC se utilizan para identificar imágenes de referencia entre capas. Es decir, cuando una imagen se codifica utilizando predicción entre capas (por ejemplo, predicción entre vistas), se señala un valor de POC de una imagen de referencia de manera que un decodificador de vídeo pueda identificar la imagen de referencia. Cuando los valores de POC están alineados, es sencillo determinar que una imagen actual se predice entre capas, concretamente, cuando el valor de POC de la imagen de referencia es el mismo que el valor de POC de la imagen actual. Los valores de POC desalineados de las imágenes entre las capas de codificación de vídeo dificultan la identificación de las imágenes de referencia. Además, los límites de la unidad de acceso se vuelven difíciles de detectar y pueden ser propensos a errores.

50

55 **[0017]** Por lo tanto, esta divulgación describe técnicas para permitir imágenes IRAP no alineadas y al mismo tiempo garantizar que los valores de POC para imágenes en una unidad de acceso común estén alineados de forma cruzada. En particular, un elemento sintáctico puede indicar que al menos una porción de un valor de POC para una imagen se debe restablecer (es decir, establecerse en un valor de cero). Por ejemplo, un codificador de vídeo puede determinar que una unidad de acceso incluye una imagen IRAP en una capa y una imagen no IRAP en otra capa diferente. Por lo tanto, el codificador de vídeo puede indicar que el valor de POC para la imagen no IRAP se debe restablecer. Un decodificador de vídeo puede restablecer el valor de POC para la imagen no IRAP, y también modificar los valores de POC de otras imágenes en la misma capa de codificación de vídeo que la imagen no IRAP. Es decir, el decodificador de vídeo puede modificar los valores de POC de estas otras imágenes, de manera que las diferencias entre el valor de POC de la imagen no IRAP y otras imágenes en la misma capa y la misma CVS permanezcan constantes después de restablecer el valor de POC de la imagen no IRAP.

60

65 **[0018]** Al ajustar los valores de POC de esta manera, se puede asegurar que todas las imágenes en una unidad de acceso común tengan el mismo valor de POC, sin embargo, las imágenes IRAP no necesitan estar alineadas. Por lo

tanto, la detección de los límites de la unidad de acceso puede simplificarse y hacerse más tolerante a los errores, y la identificación de imágenes de referencia entre capas también puede simplificarse.

5 **[0019]** Las normas de codificación de vídeo incluyen ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 o ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual e ITU-T H.264 (también conocida como ISO/IEC MPEG-4 AVC), incluyendo sus ampliaciones de codificación de vídeo escalable (SVC) y de codificación de vídeo multivista (MVC).

10 **[0020]** Recientemente, el Equipo Mixto de Colaboración sobre 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 ha terminado el diseño de un nuevo estándar de codificación de vídeo, concretamente la codificación de vídeo de alta eficacia (HEVC). El borrador de la memoria descriptiva de HEVC más reciente, denominado de aquí en adelante HEVC WD10, está disponible en http://phenix.int-evry.fr/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1003-v34.zip
 15 El JCT-3V también está desarrollando la extensión multivista de la HEVC, concretamente, la MV-HEVC. Un borrador de trabajo (WD) reciente de la MV-HEVC, denominado MV-HEVC WD3 de aquí en adelante está disponible en: http://phenix.it-sudparis.eu/jct2/doc_end_user/documents/3_Geneva/wg11/JCT3V-C1004-v4.zip
 La extensión escalable de la HEVC, concretamente la SHV, también se está desarrollando por parte del JCT-VC. Un borrador de trabajo (WD) reciente de la SHVC, denominado SHVC WD1 de aquí en adelante, está disponible en: http://phenix.int-evry.fr/jct/doc_end_user/documents/12_Geneva/wg11/JCTVC-L1008-v1.zip.

20 **[0021]** Este documento supone los siguientes conceptos de imagen (codificada), unidad de acceso (AU), IRAP AU, secuencia de vídeo codificada (CVS) y flujo de bits, aunque los procedimientos también se aplican si algunos de estos supuestos se cambian:

- 25
- imagen (codificada): Similar a la definición de imagen actual (codificada), donde una imagen codificada es equivalente a una representación de capa en SVC, un componente de la vista en MVC y un componente de textura o profundidad de la vista en MVC+ D.
 - 30 - unidad de acceso: Similar a SVC y MVC, una AU consiste en todas las imágenes codificadas asociadas con el mismo tiempo de salida y sus unidades NAL no VCL asociadas.
 - unidad de acceso IRAP: Una unidad de acceso en la que todas las imágenes codificadas son imágenes IRAP.
 - 35 - secuencia de vídeo codificada (CVS): Una secuencia de unidades de acceso que consiste, en el orden de decodificación, en una unidad de acceso IRAP con NoRasiOutputFlag igual a 1, seguida de cero o más unidades de acceso que no son unidades de acceso IRAP con NoRasiOutputFlag igual a 1, incluidas todas las unidades de acceso posteriores hasta, pero sin incluir ninguna unidad de acceso posterior que sea una unidad de acceso con NoRasiOutputFlag igual a 1.
 - 40
 - o Obsérvese que la redacción es la misma que en HEVC WD10.
 - flujo de bits: Una secuencia de bits, en forma de un flujo de unidades NAL o un flujo de bytes, que forma la representación de una o más CVS.
 - 45
 - o La primera AU en un flujo de bits será una AU IRAP (como se ha definido anteriormente).

50 **[0022]** Las imágenes de actualización instantánea del decodificador (IDR), las imágenes de acceso aleatorio limpio (CRA) y las imágenes de acceso de enlace roto (BLA) se denominan colectivamente imágenes IRAP. Requerir la alineación de capa cruzada para las imágenes IRAP no permitiría algunos escenarios de utilización ventajosos. 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 una sintonización de bajo retardo y al mismo tiempo una alta eficacia en la codificación. Por lo tanto, es deseable permitir imágenes IRAP no alineadas.

55 **[0023]** Sin embargo, cuando una imagen (picA) es una imagen IRAP con NoRasiOutputFlag igual a 1 y otra imagen (picB) en la misma unidad de acceso no es una imagen IRAP con NoRasiOutputFlag igual a 1, el valor de POC de una imagen (picC) en la capa que contiene picA puede ser diferente del valor de POC de una imagen (picD) en la capa que contiene picB, donde picC y picD están en la misma unidad de acceso, y picC puede o no ser picA (y, por consiguiente, picD puede o no ser picB). Esto se debe a que cada imagen IRAP con NoRasiOutputFlag igual a 1 restablece el valor de POC para que sea igual a los bits menos significativos de POC (LSB) señalizados o derivados para la imagen IRAP con NoRasiOutputFlag igual a 1.

60

[0024] Esto dificulta la identificación de imágenes de referencia en la codificación de vídeo multicapa con predicción entre capas en comparación con cuando los valores de POC están alineados de forma cruzada. Por ejemplo, en MV-

HEVC WD3, el POC se utiliza como una de las dos identificaciones de dimensión para identificar una imagen de referencia entre capas.

[0025] Además, esto hace que la detección de límites de la unidad de acceso (AU) sea más difícil y no tolerante a los errores. Por ejemplo, suponga que AU1 consiste en picA y picB en la capa base y la capa de mejora, respectivamente, AU2 consiste en picC y picD en la capa base y la capa de mejora, respectivamente, y AU1 precede a AU2 en el orden de descodificación. Luego, si tanto picB como picC se pierden, no hay forma de que un descodificador sepa que las picA y picD recibidas pertenecen a dos UA diferentes en base a las imágenes codificadas. En el mismo ejemplo, si AU2 consiste en solamente picD en la capa de mejora mientras que picC nunca existe, entonces se producirá el mismo problema incluso cuando solamente se pierde una imagen, picB.

[0026] Esta divulgación describe técnicas que permiten la no alineación cruzada de imágenes IRAP con NoRasOutputFlag igual a 1, mientras que al mismo tiempo proporciona una alineación cruzada de los valores de POC para todas las AU al mismo tiempo.

[0027] La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación y descodificación de vídeo de ejemplo de acuerdo con uno o más ejemplos descritos en esta divulgación. Por ejemplo, el sistema 10 incluye el dispositivo de origen 12 y el dispositivo de destino 14. El dispositivo de origen 12 y el dispositivo de destino 14 están configurados para implementar las técnicas descritas en esta divulgación. En algunos ejemplos, el sistema 10 puede configurarse para admitir la codificación, transmisión, almacenamiento, descodificación y/o presentación de datos de vídeo codificados, tales como datos de vídeo codificados de acuerdo con la norma HEVC, por ejemplo, como se describe en WD10, y sus extensiones, tales como, por ejemplo, las extensiones descritas en MV-HEVC WD3, SHVC WD1, o similares. Sin embargo, las técnicas descritas en esta divulgación pueden ser aplicables a otras normas de codificación de vídeo u otras extensiones.

[0028] Como se muestra en la FIG. 1, el sistema 10 incluye un dispositivo de origen 12 que genera datos de vídeo codificados que un dispositivo de destino 14 va a descodificar en un momento posterior. El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender cualquiera entre una amplia gama de dispositivos, incluyendo ordenadores de mesa, ordenadores plegables (es decir, portátiles), tabletas electrónicas, cajas de conexión, aparato telefónico con microteléfono tales como los denominados teléfonos "inteligentes", los denominados paneles "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, un dispositivo de emisión de vídeo en tiempo real o similares. En algunos casos, el dispositivo de origen 12 y el dispositivo de destino 14 pueden estar equipados para la comunicación inalámbrica.

[0029] El dispositivo de destino 14 puede recibir los datos de vídeo codificados que se van a descodificar por medio de un enlace 16. El enlace 16 puede comprender cualquier tipo de medio o dispositivo capaz de transferir los datos de vídeo codificados desde el dispositivo de origen 12 hasta el dispositivo de destino 14. En un ejemplo, el enlace 16 puede comprender un medio de comunicación para permitir que el dispositivo de origen 12 transmita datos de vídeo codificados directamente a un dispositivo de destino 14 en tiempo real. Los datos de vídeo codificados se pueden modular de acuerdo con una norma de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitir al dispositivo de destino 14. El medio de comunicación puede comprender cualquier medio de comunicación inalámbrica o alámbrica, tal como un espectro de radiofrecuencia (RF) o una o más líneas físicas de transmisión. El medio de comunicación puede formar parte de una red basada en paquetes, tal como una red de área local, una red de área amplia o una red global, tal como Internet. El medio de comunicación puede incluir encaminadores, conmutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo de origen 12 hasta el dispositivo de destino 14.

[0030] De forma alternativa, los datos codificados se pueden emitir desde la interfaz de salida 22 a un dispositivo de almacenamiento 34. De forma similar, una interfaz de entrada puede acceder a los datos codificados del dispositivo de almacenamiento 34. El dispositivo de almacenamiento 34 puede incluir cualquiera de una variedad de medios de almacenamiento de datos de acceso distribuido o local, tales como una unidad de disco duro, discos Blu-ray, DVD, CD-ROM, una memoria flash, una memoria volátil o no volátil o cualquier otro medio de almacenamiento digital adecuado para almacenar datos de vídeo codificados. En otro ejemplo, el dispositivo de almacenamiento 34 puede corresponder a un servidor de archivos o a otro dispositivo de almacenamiento intermedio que puede contener el vídeo codificado generado por el dispositivo de origen 12. El dispositivo de destino 14 puede acceder a datos de vídeo almacenados en el dispositivo de almacenamiento 34 a través de transmisión continua o descarga. El servidor de archivos puede ser cualquier tipo de servidor capaz de almacenar datos de vídeo codificados y transmitir esos datos de vídeo codificados al dispositivo de destino 14. Servidores de archivos de ejemplo incluyen un servidor web (por ejemplo, para un sitio web), un servidor FTP, dispositivos de almacenamiento conectados en red (NAS) o una unidad de disco local. El dispositivo de destino 14 puede acceder a los datos de vídeo codificados a través de cualquier conexión de datos estándar, incluida una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión Wi-Fi), una conexión alámbrica (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 34 puede ser una transmisión continua, una transmisión de descarga o una combinación de ambas cosas.

[0031] Las técnicas de esta divulgación no están limitadas a aplicaciones o configuraciones inalámbricas. Las técnicas se pueden aplicar a la codificación de vídeo como apoyo a cualquiera de una variedad de aplicaciones multimedia, tales como radiodifusiones de televisión por aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones continuas de vídeo, por ejemplo, a través de Internet, 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 10 puede estar configurado para prestar soporte a una transmisión de vídeo unidireccional o bidireccional, a fin de prestar soporte a aplicaciones tales como la transmisión continua de vídeo, la reproducción de vídeo, la radiodifusión de vídeo y/o la videotelefonía.

[0032] En el ejemplo de la FIG. 1, el dispositivo 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 dispositivo de origen 12, la fuente de vídeo 18 puede incluir una fuente tal como un dispositivo de captación de vídeo, por ejemplo, una videocámara, un archivo de vídeo que contiene vídeo captado previamente, una interfaz de alimentación de vídeo para recibir vídeo desde un proveedor de contenido de vídeo y/o un sistema de gráficos por ordenador para generar datos de gráficos por ordenador como el vídeo de origen, o una combinación de dichas fuentes. En un ejemplo, si la fuente de vídeo 18 es una videocámara, el dispositivo de origen 12 y el dispositivo de destino 14 pueden formar los denominados teléfonos con cámara o videoteléfonos. 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.

[0033] El vídeo captado, precaptado o generado por ordenador se puede codificar por el codificador de vídeo 20. Los datos de vídeo codificados se pueden transmitir directamente al dispositivo de destino 14 por medio de la interfaz de salida 22 del dispositivo de origen 12. Los datos de vídeo codificados se pueden almacenar de forma adicional (o alternativa) en el dispositivo de almacenamiento 34 para un posterior acceso por el dispositivo de destino 14 u otros dispositivos, para su descodificación y/o reproducción.

[0034] El dispositivo de destino 14 incluye una interfaz de entrada 28, un descodificador de vídeo 30 y un dispositivo de visualización 32. En algunos casos, la interfaz de entrada 28 puede incluir un receptor y/o un módem. La interfaz de entrada 28 del dispositivo de destino 14 recibe los datos de vídeo codificados a través del enlace 16. Los datos de vídeo codificados transmitidos a través del enlace 16, o proporcionados en el dispositivo de almacenamiento 34, pueden incluir una diversidad de elementos sintácticos generados por el codificador de vídeo 20 para su uso por un descodificador de vídeo, tal como el descodificador de vídeo 30, en la descodificación de los datos de vídeo. Dichos elementos 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.

[0035] El dispositivo de visualización 32 se puede integrar con, o ser externo a, el dispositivo de destino 14. En algunos ejemplos, el dispositivo de destino 14 puede incluir un dispositivo de visualización integrado y también estar configurado para interconectarse con un dispositivo de visualización externo. En otros ejemplos, el dispositivo de destino 14 puede ser un dispositivo de visualización. En general, el dispositivo de visualización 32 visualiza los datos de vídeo descodificados ante un usuario, y puede comprender cualquiera de una variedad de dispositivos de visualización, tales como una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodos orgánicos emisores de luz (OLED) u otro tipo de dispositivo de visualización.

[0036] 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 ITU-T H.264, denominada de forma alternativa MPEG-4, parte 10, codificación avanzada de vídeo (AVC), o extensiones de dichas normas. De forma alternativa, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con otras normas privadas o industriales, tales como la norma de codificación de vídeo de alta eficacia (HEVC) actualmente en desarrollo, así como las extensiones de la norma HEVC. 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.

[0037] Aunque no se muestra en la FIG. 1, en algunos aspectos, tanto el codificador de vídeo 20 como el descodificador de vídeo 30 pueden estar integrados con un codificador y descodificador de audio, y pueden incluir unidades MUX-DEMUX adecuadas, u otro tipo de hardware y software, para gestionar la codificación tanto de audio como de vídeo en un flujo de datos común o en flujos de datos separados. Si procede, en algunos ejemplos, las unidades MUX-DEMUX pueden ajustarse al protocolo de multiplexor ITU H.223 o a otros protocolos, tales como el protocolo de datagramas de usuario (UDP).

[0038] De acuerdo con las técnicas de esta divulgación, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden configurarse para codificar valores de elementos sintácticos indicativos de si los valores de contador del orden de imágenes (POC) de las imágenes deben reiniciarse parcial o totalmente. Un restablecimiento parcial puede corresponder al establecimiento de una porción del valor de POC, por ejemplo, solamente los bits más significativos (MSB), igual a cero. Un restablecimiento completo puede corresponder al establecimiento de todos los bits del valor de POC igual a cero. Por lo tanto, cuando el elemento sintáctico indica que un valor de POC de una imagen se debe

restablecer parcial o totalmente, el descodificador de vídeo 30 puede restablecer los bits correspondientes del valor de POC a cero.

5 **[0039]** Además, el descodificador de vídeo 30 también puede establecer bits de los valores de POC de otras imágenes en la misma capa que la imagen para la que se ha restablecido el valor de POC, de manera que las diferencias entre los valores de POC de estas otras imágenes permanezcan iguales después del restablecimiento como antes del restablecimiento. Por ejemplo, para cada una de las N otras imágenes, el descodificador de vídeo 30 puede determinar una diferencia D_i , donde i está entre 0 y $N-1$ inclusive y representa una de estas otras imágenes. En este ejemplo, el descodificador de vídeo 30 puede calcular $D_i = POC_{base} - POC_i$, donde "base" se refiere a la imagen para la cual se ha restablecido el valor de POC. El descodificador de vídeo 30 puede entonces disminuir de manera que $POC_i + D_i = POC_{base_reset}$, donde $POC_{i_disminuido}$ representa el valor de POC disminuido de la imagen i y POC_{base_reset} representa el valor restablecido de POC_{base} . En los ejemplos en los que el valor de POC de una imagen que se va a restablecer se establece en cero, el descodificador de vídeo 30 puede disminuir los valores de POC de otras imágenes en la misma CVS y la misma capa de codificación de vídeo en el valor de POC inicial (es decir, antes del restablecimiento) de la imagen. De forma alternativa, el descodificador de vídeo 30 puede establecer los valores de POC de otras imágenes en la misma CVS y la misma capa de codificación de vídeo igual a la diferencia entre el valor de POC inicial de la imagen para la cual se va a restablecer el valor de POC y el valor de POC de la otra imagen (es decir, igual a D_i).

20 **[0040]** En un conjunto de ejemplos, se añade un primer indicador a la cabecera de fragmento para especificar si los valores de POC se restablecen para que sean iguales a los bits menos significativos (LSB) de POC, es decir, los bits más significativos (MSB) de POC se restablecen en igual a 0, y si es así, los valores de POC de todas las imágenes en la misma capa que la imagen actual y en el búfer de imágenes descodificadas (DPB) se disminuyen, de manera que la diferencia entre los valores de POC de la imagen actual y cualquier imagen en la misma capa y en el DPB sigue siendo el mismo que si el primer indicador indica lo contrario.

30 **[0041]** En este conjunto de ejemplos, se puede añadir un segundo indicador a la cabecera de fragmento para especificar si los LSB de POC se restablecen para que sean iguales a 0. Cuando se usa junto con el primer indicador, el segundo indicador permite que los codificadores asignen libremente diferentes valores de los LSB de POC para imágenes en las diferentes capas. Este conjunto de ejemplos permite la no alineación cruzada de imágenes IRAP con `NoRasOutputFlag` igual a 1, al mismo tiempo que garantiza la alineación cruzada de los valores de POC para todas las AU al mismo tiempo, en el que las imágenes IRAP con `NoRasOutputFlag` igual a 1 pueden ser cualquiera de los tres tipos de imágenes IRAP, es decir, imágenes IDR, CRA y BLA.

35 **[0042]** En otro conjunto de ejemplos, se propone que, cuando las imágenes IDR no están alineadas en una unidad de acceso, si al menos una imagen en esa AU es una imagen IDR, los valores de POC de todas las imágenes en una unidad de acceso se establezcan en 0, similar a una imagen IDR. Esto permitiría la no alineación cruzada de imágenes IRAP con `NoRasOutputFlag` igual a 1 y mientras que al mismo tiempo garantizaría la alineación cruzada de los valores de POC para todas las AU al mismo tiempo, en el que las imágenes IRAP con `NoRasOutputFlag` igual a 1 son imágenes IDR. Más específicamente, lo siguiente se aplica a este conjunto de ejemplos:

1. Un indicador, concretamente, `idr_au_present_flag`, se señala en la cabecera de fragmento, por ejemplo, formando parte de los bits señalizados al comienzo de la cabecera de fragmento con `slice_reserved_flag[i]`.

45 a. Para una imagen con `nuh_layer_id` mayor que 0, el indicador igual a 1 indica que el valor de POC se restablece a 0, como se hace para una imagen IDR, el indicador igual a 0 indica que el valor de POC no se restablece.

50 2. Para una imagen IDR, este indicador se puede usar para otros fines, o se puede reservar para que sea 1, o puede estar ausente pero se infiere que es igual a 1.

55 **[0043]** Para ambos conjuntos de ejemplos, para una imagen con `nuh_layer_id` igual a 0, el indicador no tiene ningún efecto para un descodificador HEVC versión 1 (que es un descodificador 2D de una capa basado en HEVC WD10), ya que dicho descodificador ignora el(los) indicador(es).

60 **[0044]** A continuación se describen diversos ejemplos de implementaciones detalladas de estas técnicas de ejemplo. Los cambios propuestos a la norma se destacan a continuación, donde el texto en *cursiva* representa incorporaciones y [eliminado: ""] representa supresiones con respecto a la versión anterior de la norma. En general, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden estar configurados para realizar las técnicas de esta divulgación. Esta divulgación describe un "codificador de vídeo" que realiza estas técnicas. Se debe entender que el término "codificador de vídeo" puede referirse a un codificador de vídeo, tal como un codificador de vídeo 20, o un descodificador de vídeo, tal como un descodificador de vídeo 30. Asimismo, el término codificación de vídeo se podría referir a la codificación de vídeo o bien a la descodificación de vídeo. Un codificador de vídeo, tal como el codificador de vídeo 20 o el descodificador de vídeo 30, puede estar configurado para ejecutar cualquiera o todas las técnicas descritas en esta divulgación. En los casos en que se describen ciertas técnicas desde la perspectiva de un descodificador de vídeo, un codificador de vídeo puede realizar las mismas técnicas o similares (por ejemplo,

recíprocas). Del mismo modo, cuando se describen ciertas técnicas desde la perspectiva de un codificador de vídeo, un decodificador de vídeo puede realizar las mismas técnicas o similares (por ejemplo, recíprocas).

5 **[0045]** A continuación se describe un primer ejemplo de acuerdo con las técnicas de esta divulgación. Este ejemplo se denomina a continuación como Ejemplo 1. La sintaxis y la semántica a continuación, en general, se refieren a las secciones correspondientes en la HEVC WD10. Como se ha mencionado anteriormente, el resaltado indica incorporaciones propuestas y el tachado indica supresiones propuestas.

10 **Ejemplo 1**

[0046] A continuación, primero se describen la sintaxis y la semántica de la cabecera de segmento general.

Sintaxis de la cabecera de segmento general

	Descriptor
slice_segment_header() {	
...	
if(!dependent_slice_segment_flag) {	
<i>numRsvBitsUsed</i> = 0	
if(<i>nal_unit_type</i> != IDR_W_RADL && <i>nal_unit_type</i> != IDR_N_LP) {	
<i>poc_msb_reset_flag</i>	u(1)
<i>numRsvBitsUsed</i> ++	
if (<i>poc_msb_reset_flag</i>) {	
<i>poc_lsb_reset_flag</i>	u(1)
<i>numRsvBitsUsed</i> ++	
}	
}	
for(<i>i</i> = <i>numRsvBitsUsed</i> ; <i>i</i> < <i>num_extra_slice_header_bits</i> ; <i>i</i> ++)	
<i>slice_reserved_flag</i>[<i>i</i>]	u(1)
...	ue(v)
}	

15 **[0047]** A continuación se describe la semántica de la cabecera de segmento general. Como se ha mencionado anteriormente, el texto en *cursiva* indica incorporaciones propuestas y [eliminado: ""] indica supresiones propuestas. La semántica de los elementos sintácticos sin cambios puede permanecer igual que la semántica descrita en la versión anterior de la norma.

20 **[0048]** ***poc_msb_reset_flag*** igual a 1 especifica que el contador del orden de imágenes derivado para la imagen actual es igual a *slice_pic_order_cnt_lsb*. ***poc_msb_reset_flag*** igual a 0 especifica que el contador del orden de imágenes derivado para la imagen actual puede o no ser igual a *slice_pic_order_cnt_lsb*.

25 **[0049]** Cuando está presente, el valor de *poc_msb_reset_flag* será igual a 1 cuando la imagen actual no es una imagen IRAP con *NoRasIOutputFlag* igual a 1 y al menos una imagen en la unidad de acceso actual es una imagen IRAP con *NoRasIOutputFlag* igual a 1.

30 **[0050]** Cuando no está presente, se infiere que el valor de *poc_msb_reset_flag* es igual a 0.

[0051] ***poc_lsb_reset_flag*** igual a 1 especifica que el contador del orden de imágenes derivado para la imagen actual es igual a 0, ***poc_lsb_reset_flag*** igual a 0 especifica que el contador del orden de imágenes derivado para la imagen actual puede o no ser igual a 0.

35 **[0052]** Cuando está presente, el valor de *poc_lsb_reset_flag* será igual a 1 cuando la imagen actual no es una imagen IDR y al menos una imagen en la unidad de acceso actual es una imagen IDR.

[0053] Cuando no está presente, se infiere que el valor de *poc_lsb_reset_flag* es igual a 0.

40 **[0054]** Por consiguiente, el codificador de vídeo 20 puede establecer *poc_msb_reset_flag* para que tenga un valor de 1 para una imagen que no es una imagen IRAP y que está en una unidad de acceso que incluye una imagen IRAP, por ejemplo, en una capa de codificación de vídeo diferente. Del mismo modo, el decodificador de vídeo 30, tras

recibir un valor de 1 para una imagen que no es una imagen IRAP, puede restablecer el valor de MSB del valor de POC para la imagen y ajustar los valores de POC de otras imágenes en la misma capa y la misma secuencia de vídeo codificada según sea necesario.

5 **[0055]** A continuación se describe la semántica del proceso de descodificación general. El descodificador de vídeo 30 puede configurarse para descodificar datos de vídeo como se describe a continuación. El codificador de vídeo 20 puede configurarse para codificar datos de vídeo de acuerdo con un proceso recíproco al que se describe a continuación. Como se ha mencionado anteriormente, los cambios a continuación se pueden hacer en referencia a las porciones correspondientes de la HEVC WD10. El texto en cursiva indica incorporaciones propuestas y [eliminado: """] indica supresiones propuestas.

Proceso de descodificación general

15 **[0056]** La entrada a este proceso es un flujo de bits. La salida de este proceso es una lista de imágenes descodificadas.

[0057] La lista de identificadores de capa TargetDecLayerIdList, que especifica la lista de valores nuh_layer_id, en orden creciente de valores nuh_layer_id, de las unidades NAL a descodificar, se especifica de la siguiente manera:

20 - Si algunos medios externos, no especificados en esta memoria descriptiva, están disponibles para establecer TargetDecLayerIdList, TargetDecLayerIdList se establece mediante los medios externos.

- De lo contrario, si el proceso de descodificación se invoca en una prueba de conformidad del flujo de bits como se especifica en la subcláusula C.1, TargetDecLayerIdList se establece como se especifica en la subcláusula C.1.

25 - De lo contrario, TargetDecLayerIdList solo contiene un valor de ID de capa nuh que es igual a 0.

[0058] La variable TargetDecHighestTid, que identifica la subcapa temporal más alta a descodificar, se especifica de la siguiente manera:

30 - Si algunos medios externos, no especificados en esta memoria intermedia, están disponibles para establecer HighestTid, HighestTid se establece por medios externos.

- De lo contrario, si el proceso de descodificación se invoca en una prueba de conformidad del flujo de bits como se especifica en la subcláusula C.1, HighestTid se establece como se especifica en la subcláusula C.1.

35 - De lo contrario, HighestTid se establece igual a sps_max_sub_layers_minus1.

[0059] El proceso de extracción del subflujo de bits como se especifica en la cláusula 10 se aplica con el flujo de bits, HighestTid y TargetDecLayerIdList como entradas, y la salida se asigna a un flujo de bits denominado BitstreamToDecode.

[0060] Los procesos de descodificación que se especifican en el resto de esta subcláusula se aplican a cada imagen codificada, denominada la imagen actual y se denotan mediante la variable CurrPic, en BitstreamToDecode.

45 **[0061]** Dependiendo del valor de chroma_format_idc, el número de matrices de muestra de la imagen actual es de la siguiente manera:

- Si chroma_format_idc es igual a 0, la imagen actual consiste en 1 matriz de muestra S_L.

50 - De lo contrario (chroma_format_idc no es igual a 0), la imagen actual consiste en 3 matrices de muestra S_L, S_{Cb}, S_{Cr}.

55 **[0062]** El proceso de descodificación para la imagen actual toma como entradas los elementos sintácticos y variables en mayúsculas de la cláusula 7. Al interpretar la semántica de cada elemento sintáctico en cada unidad NAL, el término "el flujo de bits" (o una parte del mismo, por ejemplo, una CVS del flujo de bits) se refiere a BitstreamToDecode (o una parte del mismo).

60 **[0063]** El proceso de descodificación se especifica de manera que todos los descodificadores producirán imágenes descodificadas recortadas idénticas numéricamente. Cualquier proceso de descodificación que produzca imágenes descodificadas recortadas idénticas a las producidas por el proceso descrito en el presente documento (con el orden de salida o la sincronización de salida correcta, como se especifica) se amolda a los requisitos del proceso de descodificación de esta memoria descriptiva.

Cuando la imagen actual es una imagen IRAP, se aplica lo siguiente:

65

[0064]

- Si la imagen actual es una imagen IDR o una imagen BLA, la variable *NoRasOutputFlag* se establece igual a 1.

5 - De lo contrario (la imagen actual es una imagen CRA), se aplica lo siguiente:

- Si la imagen actual es la primera imagen en el flujo de bits en el orden de descodificación o la primera imagen que va detrás de una unidad NAL de fin de secuencia en el orden de descodificación, todas las imágenes en la misma unidad de acceso serán imágenes CRA, la variable *NoRasOutputFlag* se establece igual a 1.

10 - De lo contrario, si algunos medios externos no especificados en esta especificación están disponibles para establecer la variable *HandleCraAsBlaFlag* en un valor para la imagen actual, la variable *HandleCraAsBlaFlag* se establece igual al valor proporcionado por los medios externos y la variable *NoRasOutputFlag* se establece igual a *HandleCraAsBlaFlag*. En este caso, el valor de *HandleCraAsBlaFlag* para la imagen actual no será igual a 1 cuando haya una imagen no CRA en la unidad de acceso actual.

15 - De lo contrario, la variable *HandleCraAsBlaFlag* se establece igual a 0 y la variable *NoRasOutputFlag* se establece igual a 0.

20 Cuando la imagen actual tiene *nuh_layer_id* igual a 0, se invoca el proceso de descodificación para una imagen codificada con *nuh_layer_id* igual a 0 especificado en la subcláusula 8.1.1.

[0065] A continuación se describe la semántica para un proceso de descodificación de una imagen codificada con *nuh_layer_id* igual a 0. El descodificador de vídeo 30 puede configurarse para descodificar datos de vídeo como se describe a continuación. El codificador de vídeo 20 puede configurarse para codificar datos de vídeo de acuerdo con un proceso recíproco al que se describe a continuación.

Proceso de descodificación para una imagen codificada con *nuh_layer_id* igual a 0

30 **[0066]** Cuando la imagen actual es una imagen BLA que tiene *nal_unit_type* igual a *BLA_W_LP* o es una imagen CRA, se aplica lo siguiente:

- Si algunos medios externos no especificados en esta memoria descriptiva están disponibles para establecer la variable *UseAltCpbParamsFlag* en un valor, *UseAltCpbParamsFlag* se establece igual al valor proporcionado por los medios externos.

35

- De lo contrario, el valor de *UseAltCpbParamsFlag* se establece igual a 0.

- [eliminado: "Cuando la imagen actual es una imagen IRAP, se aplica lo siguiente:

40 - Si la imagen actual con un *nuh_layer_id* particular es una imagen IDR, una imagen BLA, la primera imagen con ese *nuh_layer_id* particular en el flujo de bits en el orden de descodificación, o la primera imagen con ese *nuh_layer_id* particular que va detrás de una unidad NAL de fin de secuencia en el orden de descodificación, la variable *NoRasOutputFlag* se establece igual a 1.

45 - De lo contrario, si algunos medios externos no especificados en esta especificación están disponibles para establecer la variable *HandleCraAsBlaFlag* en un valor para la imagen actual, la variable *HandleCraAsBlaFlag* se establece igual al valor proporcionado por los medios externos y la variable *NoRasOutputFlag* se establece igual a *HandleCraAsBlaFlag*.

50 - De lo contrario, la variable *HandleCraAsBlaFlag* se establece igual a 0 y la variable *NoRasOutputFlag* se establece igual a 0".

[0067] Dependiendo del valor de *separate_colour_plane_flag*, el proceso de descodificación se estructura como sigue:

55 - Si *separate_colour_plane_flag* es igual a 0, el proceso de descodificación se invoca una sola vez siendo la imagen actual la salida.

60 - De lo contrario (*separate_colour_plane_flag* es igual a 1), el proceso de descodificación se invoca tres veces. Las entradas al proceso de descodificación son todas las unidades NAL de la imagen codificada con el valor idéntico de *colour_plane_id*. El proceso de descodificación de unidades NAL con un valor particular de *colour_plane_id* se especifica como si solo un CVS con formato de color monocromo con ese valor particular de *colour_plane_id* estuviera presente en el flujo de bits. La salida de cada uno de los tres procesos de descodificación se asigna a una de las 3

matrices de muestra de la imagen actual, con las unidades NAL con colour_plane_id igual a 0, 1 y 2 asignadas a S_L, S_{Cb}, y S_{Cr}, respectivamente.

5 NOTA - La variable ChromaArrayType toma el valor 0 cuando separate_colour_plane_flag es igual a 1 y chroma_format_idc es igual a 3. En el proceso de descodificación, el valor de esta variable se evalúa, dando como resultado operaciones idénticas a las de las imágenes monocromo (cuando chroma_format_idc es igual a 0).

[0068] El proceso de descodificación funciona de la siguiente manera para la imagen actual CurrPic:

- 10 **1.** La descodificación de unidades NAL se especifica en la subcláusula 8,2.
- 2.** Los procesos en la subcláusula 8,3 especifican los siguientes procesos de descodificación utilizando elementos sintácticos en la capa de segmento y superiores:
- 15 - Las variables y funciones relacionadas con el contador del orden de imágenes se obtienen en la subcláusula 8.3.1. Esto debe invocarse solo para el primer segmento de una imagen.
- Se invoca el proceso de descodificación para RPS en la subcláusula 8,3,2, donde las imágenes de referencia pueden marcarse como “no utilizadas de referencia” o “utilizadas de referencia a largo plazo”. Esto debe invocarse solo para el primer segmento de una imagen.
- 20 - Cuando la imagen actual es una imagen BLA o es una imagen CRA con NoRasIOutputFlag igual a 1, se invoca el proceso de descodificación para generar imágenes de referencia no disponibles especificadas en la subcláusula 8.3.3, que solamente necesita invocarse para el primer segmento de una imagen.
- 25 - PicOutputFlag se establece de la siguiente manera:
- Si la imagen actual es una imagen RASL y NoRasIOutputFlag de la imagen IRAP asociada es igual a 1, PicOutputFlag se establece igual a 0.
- 30 - De lo contrario, PicOutputFlag se establece igual a pic_output_flag.
- Al comienzo del proceso de descodificación para cada segmento P o B, se invoca el proceso de descodificación para la construcción de listas de imágenes de referencia especificadas en la subcláusula 8.3.4 para la derivación de la lista de imágenes de referencia 0 (RefPicList0), y cuando se descodifica un segmento B, la lista de imágenes de referencia 1 (RefPicList1).
- 35 **3.** Los procesos en las subcláusulas 8.4, 8.5, 8.6 y 8.7 especifican procesos de descodificación utilizando elementos sintácticos en todas las capas de estructura sintáctica. Es un requisito de conformidad del flujo de bits que los fragmentos codificados de la imagen contengan datos de segmentos de fragmentos para cada unidad de árbol de codificación de la imagen, de manera que la división de la imagen en fragmentos, la división de los fragmentos en segmentos y la división de los segmentos en unidades de árbol de codificación formen, cada una, una partición de la imagen.
- 40 **4.** Después de que se hayan descodificado todos los segmentos de la imagen actual, la imagen descodificada se marca como “utilizada de referencia a corto plazo”.

50 **[0069]** A continuación se describe la semántica para el proceso de descodificación para el contador del orden de imágenes.

Proceso de descodificación para el contador del orden de imágenes

[0070] La salida de este proceso es PicOrderCntVal, el contador del orden de imágenes de la imagen actual.

55 **[0071]** Los contadores del orden de imágenes se usan para identificar imágenes, para derivar parámetros de movimiento en modo de combinación y predicción de vectores de movimiento, y para la verificación de conformidad del descodificador (véase la subcláusula C.5).

60 **[0072]** Cada imagen codificada está asociada a una variable de contador del orden de imágenes, denominada PicOrderCntVal.

Cuando la imagen actual no es una imagen IRAP con NoRasIOutputFlag igual a 1, las variables prevPicOrderCntLsb y prevPicOrderCntMsb se derivan de la siguiente manera:

- 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 que no es de referencia, y supongamos que prevPicOrderCnt igual a PicOrderCntVal de prevTid0Pic.

5 - La variable prevPicOrderCntLsb se establece igual a prevPicOrderCnt & (MaxPicOrderCntLsb - 1) [eliminado: "slice_pic_order_cnt_lsb of prevTid0Pic"].

10 - La variable prevPicOrderCntMsb se establece igual a prevPicOrderCnt - prevPicOrderCntLsb [eliminado: "PicOrderCntMsb of prevTid0Pic"].

[0073] La variable PicOrderCntMsb de la imagen actual se deriva de la siguiente manera:

15 - Si la imagen actual es una imagen IRAP con NoRaslOutputFlag igual a 1, PicOrderCntMsb se establece igual a 0.
- De lo contrario, PicOrderCntMsb se deriva de la siguiente manera:

```

if( ( slice_pic_order_cnt_lsb < prevPicOrderCntLsb ) &&
    ( ( prevPicOrderCntLsb - slice_pic_order_cnt_lsb ) >= (
MaxPicOrderCntLsb / 2 ) ) )
    PicOrderCntMsb = prevPicOrderCntMsb + MaxPicOrderCntLsb (8-1)
else if( ( slice_pic_order_cnt_lsb > prevPicOrderCntLsb ) &&
    ( ( slice_pic_order_cnt_lsb - prevPicOrderCntLsb ) > ( MaxPicOrderCntLsb
/ 2 ) ) )
    PicOrderCntMsb = prevPicOrderCntMsb - MaxPicOrderCntLsb
else
    PicOrderCntMsb = prevPicOrderCntMsb

```

PicOrderCntVal se deriva de la siguiente manera:

$$\text{PicOrderCntVal} = (\text{poc_msb_reset_flag} ? 0 : \text{PicOrderCntMsb}) + (8-2)$$

$$(\text{poc_lsb_reset_flag} ? 0 : \text{slice_pic_order_cnt_lsb})$$

20 **[0074]** NOTA 1 - Todas las imágenes IDR 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 a 0.

25 **[0075]** Cuando poc_msbreset_flag es igual a 1, PicOrderCntVal de cada imagen que está en el DPB y pertenece a la misma capa que la imagen actual se disminuye en PicOrderCntVal.

[0076] Cuando poc_lsb_reset_flag es igual a 1, PicOrderCntVal de cada imagen que está en el DPB y pertenece a la misma capa que la imagen actual se disminuye en slice_pic_order_cnt_lsb.

30 **[0077]** El valor de PicOrderCntVal estará en el intervalo de -2^{31} a $2^{31} - 1$, inclusives. En una CVS, los valores de PicOrderCntVal para dos imágenes codificadas cualesquiera en la misma capa no serán iguales.

[0078] La función PicOrderCnt(picX) se especifica de la siguiente manera:

35
$$\text{PicOrderCnt}(\text{picX}) = \text{PicOrderCntVal de la imagen picX (8 -3)}$$

[0079] La función DiffPicOrderCnt(picA, picB) se especifica de la siguiente manera:

$$\text{DiffPicOrderCnt}(\text{picA}, \text{picB}) = \text{PicOrderCnt}(\text{picA}) - \text{PicOrderCnt}(\text{picB}) \quad (8-4)$$

5 **[0080]** El flujo de bits no debe contener datos que den como resultado valores de $\text{DiffPicOrderCnt}(\text{picA}, \text{picB})$ usados en el proceso de descodificación que no estén en el intervalo de -2^{15} a $2^{15} - 1$, inclusive.

10 **[0081]** NOTA 2 - Sea X la imagen actual e Y y Z 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 $\text{DiffPicOrderCnt}(X, Y)$ como $\text{DiffPicOrderCnt}(X, Z)$ son positivos o ambos son negativos.

Proceso de descodificación para el conjunto de imágenes de referencia

15 **[0082]** Este proceso se invoca una vez por imagen, después de la descodificación de una cabecera de fragmento, pero antes de la descodificación de cualquier unidad de codificación y antes del proceso de descodificación para la construcción de la lista de imágenes de referencia del fragmento, como se especifica en la subcláusula 8.3.3. Este proceso puede dar como resultado que una o más imágenes de referencia en el DPB se marquen como “no utilizadas de referencia” o “utilizadas de referencia a largo plazo”.

20 **[0083]** NOTA 1 - el RPS es una descripción absoluta de las imágenes de referencia usadas en el proceso de descodificación de las imágenes codificadas actuales y futuras. La señalización de RPS es explícita en el sentido de que todas las imágenes de referencia incluidas en el RPS se enumeran explícitamente.

25 **[0084]** Una imagen descodificada en el DPB puede marcarse como “no utilizada de referencia”, “utilizada de referencia a corto plazo” o “utilizada de referencia a largo plazo”, pero solamente una de estas tres en un momento dado durante el funcionamiento del proceso de descodificación. La asignación de una de estas marcas a una imagen elimina de manera implícita otra de estas marcas cuando corresponda. Cuando se hace referencia a una imagen que está marcada como “utilizada de referencia”, esto se refiere colectivamente a la imagen marcada como “utilizada de referencia a corto plazo” o “utilizada de referencia a largo plazo” (pero no ambas).

30 **[0085]** Cuando la imagen actual es una imagen IRAP con NoRasOutputFlag igual a 1, todas las imágenes de referencia que se encuentran actualmente en el DPB (si lo hubiera) están marcadas como “no utilizadas de referencia”.

35 **[0086]** Las imágenes de referencia a corto plazo se identifican mediante sus valores PicOrderCntVal . Las imágenes de referencia a largo plazo se identifican mediante sus valores PicOrderCntVal o sus valores $\text{slice_pic_order_cnt_lsb}$.

40 **[0087]** Se construyen cinco listas de valores de contador del orden de imágenes para derivar el RPS. Estas cinco listas son PocStCurrBefore , PocStCurrAfter , PocStFoll , PocLtCurr y PocLtFoll , con el número de elementos $\text{NumPocStCurrBefore}$, NumPocStCurrAfter , NumPocStFoll , NumPocLtCurr y NumPocLtFoll , respectivamente. Las cinco listas y las cinco variables se derivan de la siguiente manera:

40 - Si la imagen actual es una imagen IDR, PocStCurrBefore , PocStCurrAfter , PocStFoll , PocLtCurr , y PocLtFoll se establecen como vacías, y $\text{NumPocStCurrBefore}$, NumPocStCurrAfter , NumPocStFoll , NumPocLtCurr , y NumPocLtFoll se establecen todos igual a 0.

45 - De lo contrario, se aplica lo siguiente:

```

    for( i = 0, j = 0, k = 0; i < NumNegativePics[ CurrRpsIdx ] ; i++ )
    if( UsedByCurrPicS0[ CurrRpsIdx ][ i ] )
        PocStCurrBefore[ j++ ] = PicOrderCntVal + DeltaPocS0[ CurrRpsIdx ][ i ]
    else
        PocStFoll[ k++ ] = PicOrderCntVal + DeltaPocS0[ CurrRpsIdx ][ i ]
    NumPocStCurrBefore = j

```

```

    for( i = 0, j = 0; i < NumPositivePics[ CurrRpsIdx ]; i++ )
    if( UsedByCurrPicS1[ CurrRpsIdx ][ i ] )
        PocStCurrAfter[ j++ ] = PicOrderCntVal + DeltaPocS1[ CurrRpsIdx ][ i ]
    else
        PocStFoll[ k++ ] = PicOrderCntVal + DeltaPocS1[ CurrRpsIdx ][ i ]
    NumPocStCurrAfter = j

```

NumPocStFoll = k (8-5)

```

    for( i = 0, j = 0, k = 0; i < num_long_term_sps + num_long_term_pics; i++ ) {
        pocLt = PocLsbLt[ i ]
        if( delta_poc_msb_present_flag[ i ] )
            pocLt += PicOrderCntVal - DeltaPocMsbCycleLt[ i ] *
            MaxPicOrderCntLsb -
                PicOrderCntVal & ( MaxPicOrderCntLsb - 1 )
        if( UsedByCurrPicLt[ i ] ) {
            PocLtCurr[ j ] = pocLt
            CurrDeltaPocMsbPresentFlag[ j++ ] = delta_poc_msb_present_flag[ i ]
        } else {
            PocLtFoll[ k ] = pocLt
            FollDeltaPocMsbPresentFlag[ k++ ] = delta_poc_msb_present_flag[ i ]
        }
    }
    NumPocLtCurr = j
    NumPocLtFoll = k

```

donde PicOrderCntVal es el contador del orden de imágenes de la imagen actual como se especifica en la subcláusula 8.3.1.

[0088] NOTA 2 - Un valor de CurrRpsIdx en el intervalo de 0 a num_short_term_ref_pic_sets - 1, inclusive, indica que se está utilizando un RPS candidato a corto plazo del SPS activo, donde CurrRpsIdx es el índice del RPS candidato a corto plazo en la lista de candidatos RPS a corto plazo señalizados en el SPS activo. CurrRpsIdx igual a num_short_term_ref_pic_sets indica que el RPS a corto plazo de la imagen actual se señala directamente en la cabecera de fragmento.

[0089] Para cada i en el intervalo de 0 a NumPocLtCurr - 1, inclusive, cuando CurrDeltaPocMsbPresentFlag[i] es igual a 1, es un requisito de conformidad del flujo de bits que se apliquen las siguientes condiciones:

Ejemplo 2

[0090] La siguiente descripción se refiere a un segundo ejemplo de acuerdo con esta divulgación. La sintaxis y la semántica a continuación, en general, se refieren a las secciones correspondientes en la HEVC WD10. Como se ha mencionado anteriormente, el texto en cursiva indica incorporaciones propuestas y [eliminado: ""] indica supresiones propuestas.

[0091] A continuación, primero se describen la sintaxis y la semántica de la cabecera de segmento general.

Sintaxis de la cabecera de segmento general

	Descriptor
slice_segment_header() {	
...	
if(!dependent_slice_segment_flag) {	
<i>poc_reset_flag</i>	u(1)
for (i = 1 [eliminado: "0"]; i < num_extra_slice_header_bits; i++)	
slice_reserved_flag [i]	u(1)
...	
}	

[0092] De forma alternativa, la sintaxis puede ser de la siguiente manera:

	Descriptor
slice_segment_header() {	
...	
if(!dependent_slice_segment_flag) {	
<i>numRsvBitsUsed = 0</i>	
<i>if(nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) {</i>	
<i>poc_reset_flag</i>	u(1)
<i>numRsvBitsUsed = 1</i>	
<i>}</i>	
for(i = <i>numRsvBitsUsed</i> [eliminado: "0"]; i < num_extra_slice_header_bits; i++)	
slice_reserved_flag [i]	u(1)
...	ue(v)
}	

[0093] A continuación se describe la semántica de la cabecera de segmento general.

Semántica de la cabecera de segmento general

[0094] *poc_reset_flag* igual a 1 especifica que el contador del orden de imágenes derivado para la imagen actual es igual a 0, *poc_lsb_reset_flag* igual a 0 especifica que el contador del orden de imágenes derivado para la imagen actual puede o no ser igual a 0.

[0095] Cuando está presente, el valor de *poc_reset_flag* será igual a 1 cuando la imagen actual no es una imagen IDR y al menos una imagen en la unidad de acceso actual es una imagen IDR.

[0096] Cuando no está presente, se infiere que el valor de *poc_reset_flag* es igual a 0.

[0097] Por consiguiente, el codificador de vídeo 20 puede establecer *poc_reset_flag* para que tenga un valor de 1 para una imagen que no es una imagen IRAP y que está en una unidad de acceso que incluye una imagen IRAP, por ejemplo, en una capa de codificación de vídeo diferente. Del mismo modo, el decodificador de vídeo 30, tras recibir un valor de 1 para una imagen que no es una imagen IRAP, puede restablecer el valor de POC de la imagen y ajustar los valores de POC de otras imágenes en la misma capa y la misma secuencia de vídeo codificada según sea necesario.

[0098] Los cambios en el proceso de decodificación propuestos de acuerdo con este segundo ejemplo se describen a continuación.

Cambios en el proceso de decodificación

Proceso de decodificación de fragmentos

8.3.1 Proceso de decodificación para el contador del orden de imágenes

[0099] La salida de este proceso es *PicOrderCntVal*, el contador del orden de imágenes de la imagen actual. Los contadores del orden de imágenes se usan para identificar imágenes, para derivar parámetros de movimiento en modo de combinación y predicción de vectores de movimiento, y para la verificación de conformidad del decodificador (véase la subcláusula C.5).

Cada imagen codificada está asociada a una variable de contador del orden de imágenes, denominada *PicOrderCntVal*.

Cuando la imagen actual no es una imagen IRAP con *NoRasIOutputFlag* igual a 1, las variables *prevPicOrderCntLsb* y *prevPicOrderCntMsb* se derivan de la siguiente manera:

- 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 que no es de referencia, y supongamos que *prevPicOrderCnt* igual a *PicOrderCntVal* de *prevTid0Pic*.

- La variable *prevPicOrderCntLsb* se establece igual a *prevPicOrderCnt & (MaxPicOrderCntLsb - 1)* [eliminado: "slice_pic_order_cnt_lsb of *prevTid0Pic*"].

- La variable *prevPicOrderCntMsb* se establece igual a *prevPicOrderCnt - prevPicOrderCntLsb* [eliminado: "*PicOrderCntMsb* of *prevTid0Pic*"].

[0100] La variable *PicOrderCntMsb* de la imagen actual se deriva de la siguiente manera:

- Si la imagen actual es una imagen IRAP con *NoRasIOutputFlag* igual a 1, *PicOrderCntMsb* se establece igual a 0.

- De lo contrario, *PicOrderCntMsb* se deriva de la siguiente manera:

```

if( ( slice_pic_order_cnt_lsb < prevPicOrderCntLsb ) &&
    ( ( prevPicOrderCntLsb - slice_pic_order_cnt_lsb ) >= (
MaxPicOrderCntLsb / 2 ) ) )
    PicOrderCntMsb = prevPicOrderCntMsb + MaxPicOrderCntLsb (8-1)
else if( (slice_pic_order_cnt_lsb > prevPicOrderCntLsb ) &&
    ( (slice_pic_order_cnt_lsb - prevPicOrderCntLsb ) > ( MaxPicOrderCntLsb
/ 2 ) ) )
    PicOrderCntMsb = prevPicOrderCntMsb - MaxPicOrderCntLsb
else
    PicOrderCntMsb = prevPicOrderCntMsb

```

PicOrderCntVal se deriva de la siguiente manera:

$$\text{PicOrderCntVal} = \text{PicOrderCntMsb} + \text{slice_pic_order_cnt_lsb} \quad (8-2)$$

5 **[0101]** NOTA 1 - Todas las imágenes IDR 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 a 0.

[0102] Cuando poc_reset_flag es igual a 1, se aplica lo siguiente en orden:

10 - El PicOrderCntVal de cada imagen que está en el DPB y pertenece a la misma capa que la imagen actual se disminuye en PicOrderCntVal.

- PicOrderCntVal se establece en 0.

15 **[0103]** El valor de PicOrderCntVal estará en el intervalo de -2^{31} a $2^{31} - 1$, inclusivos. En una CVS, los valores de PicOrderCntVal para dos imágenes codificadas cualesquiera en la misma capa no serán iguales.

[0104] La función PicOrderCnt(picX) se especifica de la siguiente manera:

$$\text{PicOrderCnt}(\text{picX}) = \text{PicOrderCntVal de la imagen picX} \quad (8-3)$$

20 **[0105]** La función DiffPicOrderCnt(picA, picB) se especifica de la siguiente manera:

$$\text{DiffPicOrderCnt}(\text{picA}, \text{picB}) = \text{PicOrderCnt}(\text{picA}) - \text{PicOrderCnt}(\text{picB}) \quad (8-4)$$

25 **[0106]** 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 -2^{15} a $2^{15} - 1$, inclusive.

30 **[0107]** NOTA 2 - Sea X la imagen actual e Y y Z 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.

[0108] El proceso de decodificación para el conjunto de imágenes de referencia en este ejemplo puede ser el mismo que el descrito para el Ejemplo 1.

35 **[0109]** De esta manera, el codificador de vídeo 20 puede configurarse para establecer un valor para un elemento sintáctico (por ejemplo, poc_msb_reset_flag o poc_reset_flag) que indica si se debe restablecer al menos una porción de un valor de POC para una imagen. Como se ha analizado anteriormente, el codificador de vídeo 20 puede configurarse para establecer el valor del elemento sintáctico para indicar que el valor de POC se debe restablecer cuando la imagen se incluye en una unidad de acceso que incluye al menos una imagen IRAP. Del mismo modo, el
40 descodificador de vídeo 30 puede determinar a partir del valor del elemento sintáctico si restablecer al menos una porción de un valor de POC (o el valor de POC completo) de una imagen. Por ejemplo, donde el elemento sintáctico es el indicador poc_msb_reset, el descodificador de vídeo 30 puede restablecer los MSB del valor de POC de la

imagen. Cuando el elemento sintáctico es `poc_reset_flag`, el descodificador de vídeo 30 puede restablecer el valor de POC completo de la imagen.

5 **[0110]** Además, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden ajustar los valores de POC de otras imágenes en la misma capa y en la misma secuencia de vídeo codificada (CVS), de manera que las diferencias entre los valores de POC en relación con el valor de POC a ajustar permanezcan constantes antes de restablecer y después. De esta manera, cuando el codificador de vídeo 20 codifica un bloque de una imagen usando la predicción entre capas, donde la imagen (después del ajuste POC analizado anteriormente) tiene un valor de POC de N, el bloque puede identificar la imagen de referencia usando el valor de POC de N. Es decir, después del ajuste, cada
10 imagen en una unidad de acceso tendrá el mismo valor de POC.

[0111] Además, el descodificador de vídeo 30 puede configurarse para detectar límites de unidades de acceso.

15 **[0112]** Tanto el codificador de vídeo 20 como el descodificador de vídeo 30 se pueden implementar como cualquiera de una variedad de circuitos codificadores adecuados, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), matrices de puertas programables *in situ* (FPGA), lógica discreta, software, hardware, firmware o cualquier combinación de los mismos. Cuando las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio adecuado no transitorio legible por ordenador, y ejecutar las instrucciones en hardware usando uno o más
20 procesadores para realizar las técnicas de esta divulgación. Tanto el codificador de vídeo 20 como 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 dispositivo respectivo.

25 **[0113]** El codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar, en general, de conformidad con HEVC WD10, MV-HEVC WD3, y/o SHVC WD1, como se ha descrito anteriormente, o con otras normas o extensiones similares en los que las técnicas descritas en esta divulgación pueden ser útiles. La norma HEVC especifica varias capacidades adicionales de los dispositivos de codificación de vídeo respecto a dispositivos existentes de acuerdo con, por ejemplo, la ITU-T H.264/AVC. Por ejemplo, mientras que la norma H.264 proporciona nueve modos de codificación mediante intrapredicción, la norma HEVC puede proporcionar hasta treinta y tres modos de codificación
30 mediante intrapredicción.

[0114] En general, una trama o imagen de vídeo puede dividirse en una secuencia de bloques de árbol o unidades de codificación de mayor tamaño (LCU), que incluyen muestras tanto de luma como de croma. Un bloque de árbol en el proceso de codificación de la HEVC tiene un fin similar al de un macrobloque de la norma H.264. Un fragmento incluye un número de bloques de árbol consecutivos en orden de codificación. Una trama o imagen de vídeo puede dividirse en uno o más fragmentos. Cada bloque de árbol puede dividirse en unidades de codificación (CU) de acuerdo con un árbol cuaternario. Por ejemplo, un bloque de árbol, tal como un nodo raíz del árbol cuaternario, puede dividirse en cuatro nodos hijo, y cada nodo hijo puede, a su vez, ser un nodo padre y dividirse en otros cuatro nodos hijo. Un
35 nodo hijo final, no dividido, tal como un nodo hoja del árbol cuaternario, comprende un nodo de codificación, es decir, un bloque de vídeo codificado. Los datos sintácticos asociados a un flujo de bits codificado pueden definir un número máximo de veces que puede dividirse un bloque de árbol, y también pueden definir un tamaño mínimo de los nodos de codificación.

45 **[0115]** Una CU incluye un nodo de codificación y unidades de predicción (PU) y unidades de transformada (TU) asociadas al nodo de codificación. Un tamaño de la CU corresponde a un tamaño del nodo de codificación y debe ser de forma cuadrada. El tamaño de la CU puede variar desde 8x8 píxeles hasta el tamaño del bloque de árbol, con un máximo de 64x64 píxeles o más. Cada CU puede contener una o más PU y una o más TU. Los datos sintácticos asociados a una CU pueden describir, por ejemplo, la división de la CU en una o más PU. Los modos de división pueden diferir entre si la CU está codificada en modo directo o de salto, codificada en modo de intrapredicción o
50 codificada en modo de interpredicción. Las PU se pueden dividir para tener una forma no cuadrada. Los datos sintácticos asociados a una CU también pueden describir, por ejemplo, la división de la CU en una o más TU de acuerdo con un árbol cuaternario. Una TU puede tener una forma cuadrada o no cuadrada.

55 **[0116]** La norma HEVC admite transformaciones de acuerdo con las TU, que pueden ser diferentes para CU diferentes. El tamaño de las TU se basa típicamente en el tamaño de las PU dentro de una CU dada definida para una LCU dividida, aunque puede que no sea siempre así. Las TU son típicamente del mismo tamaño o de un tamaño más pequeño que las PU. En algunos ejemplos, las muestras residuales correspondientes a una CU pueden subdividirse en unidades más pequeñas usando una estructura de árbol cuaternario conocida como "árbol cuaternario residual" (RQT). Los nodos hoja del RQT pueden denominarse unidades de transformada (TU). Los valores de
60 diferencias de píxeles asociados a las TU pueden transformarse para producir coeficientes de transformada, que pueden cuantificarse.

[0117] En general, una PU incluye datos relacionados con el procedimiento de predicción. Por ejemplo, cuando la PU se codifica en intramodo, la PU puede incluir datos que describen un modo de intrapredicción para la PU. Para
65 mencionar otro ejemplo, cuando la PU se codifica en intermodo, la PU puede incluir datos que definen un vector de movimiento para la PU. Los datos que definen el vector de movimiento para una PU pueden describir, por ejemplo,

un componente horizontal del vector de movimiento, un componente vertical del vector de movimiento, una resolución para el vector de movimiento (por ejemplo, precisión de píxeles de un cuarto o precisión de píxeles de un octavo), una imagen de referencia a la que apunta el vector de movimiento y/o una lista de imágenes de referencia (por ejemplo, RefPicList0 (L0) o RefPicList1 (L1)) para el vector de movimiento.

5 **[0118]** En general, se usa una TU para los procedimientos de transformada y cuantificación. Una CU dada que tiene una o más PU también puede incluir una o más unidades de transformada (TU). Después de la predicción, el codificador de vídeo 20 puede calcular valores residuales correspondientes a la PU. Los valores residuales comprenden valores de diferencias de píxeles que se pueden transformar en coeficientes de transformada, cuantificar y explorar usando las TU, para producir coeficientes de transformada en serie para la codificación por entropía. Esta divulgación usa típicamente el término “bloque de vídeo” para referirse a un nodo de codificación de una CU. En algunos casos específicos, esta divulgación también puede usar el término “bloque de vídeo” para referirse a un bloque de árbol, es decir, una LCU o una CU, que incluye un nodo de codificación y unas PU y TU.

15 **[0119]** Por ejemplo, para la codificación de vídeo de acuerdo con la norma HEVC, una trama de vídeo puede dividirse en unidades de codificación (CU), unidades de predicción (PU) y unidades de transformada (TU). Una CU en general se refiere a una región de la imagen que sirve como una unidad básica sobre la cual se aplican diversas herramientas de codificación para la compresión de vídeo. Una CU tiene típicamente una geometría cuadrada, y puede considerarse similar a un llamado “macrobloque”, en virtud de otras normas de codificación de vídeo, tales como, por ejemplo, la ITU-T H.264.

20 **[0120]** Para lograr una mejor eficacia en la codificación, una CU puede tener un tamaño variable dependiendo de los datos de vídeo que contiene. Es decir, una CU se puede partir, o “dividir” en bloques más pequeños, o sub-CU, cada uno de los cuales también se puede denominar una CU. Además, cada CU que no está dividida en sub-CU puede dividirse en una o más PU y TU con fines de predicción y transformada de la CU, respectivamente.

25 **[0121]** Las PU pueden considerarse similares a las llamadas particiones de un bloque bajo otras normas de codificación de vídeo, tales como la H.264. Las PU son la base sobre la cual se realiza la predicción para el bloque para producir coeficientes “residuales”. Los coeficientes residuales de una CU representan una diferencia entre los datos de vídeo de la CU y los datos predictivos para la CU determinados usando una o más PU de la CU. Específicamente, las una o más PU especifican cómo se divide la CU con el propósito de predicción y qué modo de predicción se utiliza para predecir los datos de vídeo contenidos dentro de cada partición de la CU.

30 **[0122]** Una o más TU de una CU especifican particiones de un bloque de coeficientes residuales de la CU sobre la base de qué transformada se aplica al bloque para producir un bloque de coeficientes de transformada residuales para la CU. Las una o más TU también pueden asociarse con el tipo de transformada que se aplica. La transformada convierte los coeficientes residuales de un dominio de píxel o espacial, a un dominio de transformada, tal como un dominio de la frecuencia. Además, las una o más TU pueden especificar parámetros sobre la base de qué cuantificación se aplica al bloque resultante de coeficientes de transformada residuales para producir un bloque de coeficientes de transformada residuales cuantificados. Los coeficientes de transformada residuales pueden cuantificarse para posiblemente reducir la cantidad de datos utilizados para representar los coeficientes.

35 **[0123]** Una CU en general incluye un componente de luminancia, denominado Y, y dos componentes de crominancia, denominados U y V. En otras palabras, una CU dada que no está dividida en sub-CU puede incluir componentes Y, U y V, cada una de las cuales se puede dividir adicionalmente en una o más PU y TU con fines de predicción y transformada de la CU, como se ha descrito anteriormente. Por ejemplo, dependiendo del formato de muestreo de vídeo, el tamaño de los componentes U y V, en términos del número de muestras, puede ser el mismo o diferente al tamaño del componente Y. Como tales, las técnicas descritas anteriormente con referencia a predicción, transformada y cuantificación pueden realizarse para cada uno de los componentes Y, U y V de una CU dada.

40 **[0124]** Para codificar una CU, se derivan primero uno o más predictores para la CU basándose en una o más PU de la CU. Un predictor es un bloque de referencia que contiene datos predictivos para la CU, y se deriva sobre la base de una PU correspondiente para la CU, como se ha descrito anteriormente. Por ejemplo, la PU indica una partición de la CU para la que se tienen que determinar los datos predictivos, y un modo de predicción utilizado para determinar los datos predictivos. Puede derivarse el predictor, ya sea mediante los modos de intrapredicción (I) (es decir, predicción espacial) o la interpredicción (P o B) (es decir, predicción temporal). Por lo tanto, algunas CU pueden estar intracodificadas (I) usando la predicción espacial con respecto a los bloques de referencia vecinos, o CU, en la misma trama, mientras que otras CU pueden estar intercodificadas (P o B) con respecto a los bloques de referencia, o CU, de otras tramas.

45 **[0125]** Tras la identificación de los uno o más predictores basados en las una o más PU de la CU, se calcula una diferencia entre los datos de vídeo originales de la CU correspondientes a las una o más PU y los datos predictivos para la CU contenidos en los uno o más predictores. Esta diferencia, también denominada valor residual de la predicción, comprende coeficientes residuales y se refiere a diferencias de píxeles entre porciones de la CU especificadas por las una o más PU y los uno o más predictores, como se ha descrito anteriormente. Los coeficientes residuales se disponen en general en una matriz bidimensional (2-D) que corresponde a las una o más PU de la CU.

[0126] Para lograr una mejor compresión, el valor residual de la predicción en general se transforma, por ejemplo, usando una transformada discreta del coseno (DCT), una transformada entera, una transformada de Karhunen-Loeve (KL) u otra transformada. La transformada convierte el valor residual de la predicción, es decir, los coeficientes residuales, en el dominio espacial en coeficientes de transformada residuales en el dominio de transformada, por ejemplo, un dominio de la frecuencia, como también se ha descrito previamente. Los coeficientes de transformada también se disponen en general en una matriz 2-D que corresponde a las una o más TU de la CU. Para una compresión adicional, los coeficientes de transformada residuales pueden cuantificarse para posiblemente reducir la cantidad de datos utilizados para representar los coeficientes, como también se ha descrito previamente.

[0127] Para lograr una compresión aún mayor, un codificador por entropía codifica posteriormente los coeficientes de transformada residuales resultantes, utilizando codificación aritmética binaria adaptativa al contexto (CABAC), codificación de longitud variable adaptativa al contexto (CAVLC), codificación por entropía por división de intervalos de probabilidad (PIPE) u otra metodología de codificación por entropía. La codificación por entropía puede conseguir esta compresión adicional reduciendo o eliminando la redundancia estadística intrínseca en los datos de vídeo de la CU, representada por los coeficientes, con relación a otras CU.

[0128] Una secuencia de vídeo incluye típicamente una serie de tramas o imágenes de vídeo. Un grupo de imágenes (GOP) comprende, en general, una serie de una o más de las imágenes de vídeo. Un GOP puede incluir datos sintácticos en una cabecera del GOP, una cabecera de una o más de las imágenes, o en otras ubicaciones, que describen una serie de imágenes incluidas en el GOP. Cada fragmento de una imagen puede incluir datos sintácticos de fragmento que describen un modo de codificación para el fragmento respectivo. El codificador de vídeo 20 actúa típicamente en bloques de vídeo dentro de fragmentos de vídeo individuales a fin de codificar los datos de vídeo. Un bloque de vídeo puede corresponder a un nodo de codificación dentro de una CU. Los bloques de vídeo pueden tener tamaños fijos o variables y pueden diferir en tamaño de acuerdo con una norma de codificación especificada.

[0129] En un ejemplo, la HEVC admite la predicción en diversos tamaños de PU. Si se supone que el tamaño de una CU particular es $2N \times 2N$, la HEVC admite la intrapredicción en tamaños de PU de $2N \times 2N$ o $N \times N$, así como la interpredicción en tamaños de PU simétricos de $2N \times 2N$, $2N \times N$, $N \times 2N$ o $N \times N$. La HEVC también admite la partición asimétrica para la interpredicción en tamaños de PU de $2N \times nU$, $2N \times nD$, $nL \times 2N$ y $nR \times 2N$. En la división asimétrica, una dirección de una CU no está dividida, mientras que la otra dirección está dividida entre un 25 % y un 75 %. La parte de la CU correspondiente a la división de 25 % está indicada por una "n" seguida de una indicación "arriba", "abajo", "izquierda" o "derecha". Así, por ejemplo, "2NxnU" se refiere a una CU $2N \times 2N$ que está dividida horizontalmente con una PU $2N \times 0,5N$ encima y una PU $2N \times 1,5N$ debajo.

[0130] En esta divulgación, "NxN" y "N por N" pueden usarse indistintamente para hacer referencia a las dimensiones de píxeles de un bloque vídeo en términos de dimensiones verticales y horizontales, por ejemplo, 16×16 píxeles o 16 por 16 píxeles. En general, un bloque de 16×16 tendrá 16 píxeles en una dirección vertical ($y = 16$) y 16 píxeles en una dirección horizontal ($x = 16$). Asimismo, un bloque de tamaño $N \times N$ tiene, en general, N píxeles en una dirección vertical y N píxeles en una dirección horizontal, donde N representa un valor entero no negativo. Los píxeles de un bloque se pueden disponer en filas y columnas. Además, no es necesario que los bloques tengan necesariamente el mismo número de píxeles en la dirección horizontal que en la dirección vertical. Por ejemplo, los bloques pueden comprender $N \times M$ píxeles, donde M no es necesariamente igual a N.

[0131] Después de la codificación de intrapredicción o interpredicción mediante las PU de una CU, el codificador de vídeo 20 puede calcular datos residuales para las TU de la CU. Las PU pueden comprender datos de píxeles en el dominio espacial (también denominado dominio del píxel) y las TU pueden comprender coeficientes en el dominio de transformada después de la aplicación de una transformada, por ejemplo, una transformada discreta del coseno (DCT), una transformada entera, una transformada de ondículas o una transformada conceptualmente similar, a los datos de vídeo residuales. Los datos residuales pueden corresponder a diferencias de píxeles entre píxeles de la imagen no codificada y valores de predicción correspondientes a las PU. El codificador de vídeo 20 puede formar las TU, incluyendo los datos residuales para la CU y, a continuación, transformar las TU para producir coeficientes de transformada para la CU.

[0132] Después de cualquier transformada para producir coeficientes de transformada, el codificador de vídeo 20 puede realizar la cuantificación de los coeficientes de transformada. La cuantificación se refiere, en general, a un proceso en el que unos coeficientes de transformada se cuantifican para reducir posiblemente la cantidad de datos usados para representar los coeficientes, proporcionando una compresión adicional. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos, o a la totalidad, de los coeficientes. Por ejemplo, un valor de n bits se puede redondear a la baja hasta un valor de m bits durante la cuantificación, donde n es mayor que m.

[0133] En algunos ejemplos, el codificador de vídeo 20 puede utilizar un orden de exploración predefinido para explorar los coeficientes de transformada cuantificados, para producir un vector en serie que se pueda codificar por entropía. En otros ejemplos, el codificador de vídeo 20 puede realizar una exploración adaptativa. Después de explorar los coeficientes de transformada cuantificados para formar un vector unidimensional, el codificador de vídeo 20 puede realizar la codificación por entropía del vector unidimensional, por ejemplo, de acuerdo con la codificación de longitud

variable adaptativa al contexto (CAVLC), la codificación aritmética binaria adaptativa al contexto (CABAC), la codificación aritmética binaria adaptativa al contexto basada en la sintaxis (SBAC), la codificación por entropía por división de intervalos de probabilidad (PIPE) u otra metodología de codificación por entropía. El codificador de vídeo 20 también puede realizar la codificación por entropía de los elementos sintácticos asociados a los datos de vídeo codificados, para su uso por el descodificador de vídeo 30 en la descodificación de los datos de vídeo.

[0134] Para realizar la CABAC, el codificador de vídeo 20 puede asignar un contexto, dentro de un modelo de contexto, a un símbolo que se va a transmitir. El contexto se puede referir, por ejemplo, a si los valores vecinos del símbolo son distintos de cero o no. Para realizar la codificación CAVLC, el codificador de vídeo 20 puede seleccionar un código de longitud variable para un símbolo a transmitir. Las palabras de código en la VLC se pueden construir de modo que los códigos relativamente más cortos correspondan a símbolos más probables, mientras que los códigos más largos correspondan a símbolos menos probables. De esta forma, el uso de la VLC puede lograr un ahorro en bits con respecto, por ejemplo, al uso de palabras de código de igual longitud para cada símbolo que se va a transmitir. La determinación de la probabilidad se puede basar en un contexto asignado al símbolo.

[0135] El codificador de vídeo 20 y el descodificador de vídeo 30 pueden estar configurados para implementar la técnicas de ejemplo de esta divulgación. El codificador de vídeo 20 y el descodificador de vídeo 30 pueden denominarse comúnmente un codificador de vídeo.

[0136] Por ejemplo, el descodificador de vídeo 30 puede configurarse para recibir un primer elemento sintáctico que especifica si un valor de contador del orden de imágenes (POC) de una imagen actual se restablece para que sea igual a un conjunto de los bits menos significativos (LSB) del valor de POC de la imagen actual, cuando el primer elemento sintáctico especifica que el valor de POC de la imagen actual se restablece para que sea igual al conjunto de LSB del valor de POC de la imagen actual, disminuya los valores de POC de otras imágenes en una misma capa de codificación como la imagen actual de modo que la diferencia entre el valor de POC de la imagen actual y el valor de POC de cualquiera de las imágenes en la misma capa permanezca igual que antes de que se restableciera el valor de POC de la imagen actual, y descodifique al menos parte de imagen actual y las otras imágenes usando los valores de POC disminuidos. Las otras imágenes pueden incluir imágenes de referencia a corto plazo e imágenes de referencia a largo plazo, y el descodificador de vídeo 30 puede disminuir los valores de POC para las imágenes de referencia a corto plazo y para las imágenes de referencia a largo plazo.

[0137] Cuando el primer elemento sintáctico especifica que el valor de POC de la imagen actual se restablece para que sea igual al conjunto de LSB del valor de POC, el descodificador de vídeo 30 puede disminuir los valores de POC de otras imágenes en un búfer de imágenes descodificadas (DPB) de modo que la diferencia entre el valor de POC de la imagen actual y el valor de POC de cualquiera de las imágenes en el DPB permanece igual que antes de que se restableciera el valor de POC de la imagen actual.

[0138] Cuando el primer elemento sintáctico especifica que el valor de POC de la imagen actual se restablece para que sea igual al conjunto de LSB del valor de POC de la imagen actual, los bits más significativos (MSB) del valor de POC de la imagen actual se restablecen a cero. En un ejemplo, los valores LSB y MSB del valor de POC de la imagen actual son mutuamente excluyentes.

[0139] En algunos ejemplos, el descodificador de vídeo 30 puede recibir un segundo elemento sintáctico que especifica si los valores LSB del valor de POC de la imagen actual se restablecen a cero. El primer y/o segundo elemento sintáctico puede ser cada uno un indicador, y uno o ambos elementos sintácticos pueden recibirse en una cabecera de fragmento. La imagen actual puede comprender una de entre una imagen IDR, CRA o BLA.

[0140] En otro ejemplo, el descodificador de vídeo 30 puede configurarse para recibir un primer elemento sintáctico que especifica si un valor de contador del orden de imágenes (POC) de una imagen actual se restablece para que sea igual a cero, cuando el primer elemento sintáctico especifica que el valor de POC de la imagen actual se restablece para que sea igual a cero, si la imagen actual está en una capa mayor que la capa 0, restablecer la imagen actual para que sea igual a cero y descodificar la imagen actual utilizando el valor de restablecimiento de POC.

[0141] En un ejemplo, la imagen actual es una imagen no IDR. El elemento sintáctico puede comprender un indicador, como un `idr_au_present_flag`. El elemento sintáctico se puede recibir en una cabecera de fragmento. Cuando la imagen actual es una imagen IDR, el elemento sintáctico siempre especifica que el valor de POC de la imagen actual se restablece para que sea igual a cero.

[0142] El codificador de vídeo 20 también puede configurarse para realizar procedimientos de acuerdo con los ejemplos de esta divulgación. Por ejemplo, el codificador de vídeo 20 puede configurarse para generar un primer elemento sintáctico que especifica si un valor de contador del orden de imágenes (POC) de una imagen actual se restablece para que sea igual a un conjunto de los bits menos significativos (LSB) del valor de POC de la imagen actual, cuando el primer elemento sintáctico especifica que el valor de POC de la imagen actual se restablece para que sea igual al conjunto de LSB del valor de POC de la imagen actual, disminuya los valores de POC de otras imágenes en una misma capa de codificación como la imagen actual de modo que la diferencia entre el valor de POC de la imagen actual y el valor de POC de cualquiera de las imágenes en la misma capa permanezca igual que antes

de que se restableciera el valor de POC de la imagen actual, y codifique al menos parte de imagen actual y las otras imágenes usando los valores de POC disminuidos.

5 **[0143]** Cuando el primer elemento sintáctico especifica que el valor de POC de la imagen actual se restablece para que sea igual al conjunto de LSB del valor de POC, el codificador de vídeo 20 puede disminuir los valores de POC de otras imágenes en un búfer de imágenes descodificadas (DPB) de modo que la diferencia entre el valor de POC de la imagen actual y el valor de POC de cualquiera de las imágenes en el DPB permanece igual que antes de que se restableciera el valor de POC de la imagen actual.

10 **[0144]** Cuando el primer elemento sintáctico especifica que el valor de POC de la imagen actual se restablece para que sea igual al conjunto de LSB del valor de POC de la imagen actual, los bits más significativos (MSB) del valor de POC de la imagen actual se restablecen a cero. En un ejemplo, los valores LSB y MSB del valor de POC de la imagen actual son mutuamente excluyentes.

15 **[0145]** El codificador de vídeo 20 puede generar un segundo elemento sintáctico que especifica si los valores LSB del valor de POC de la imagen actual se restablecen a cero. El primer y/o segundo elemento sintáctico puede ser cada uno un indicador, y uno o ambos elementos sintácticos pueden recibirse en una cabecera de fragmento. La imagen actual puede comprender una de entre una imagen IDR, CRA o BLA.

20 **[0146]** En otro ejemplo, el codificador de vídeo 20 puede configurarse para generar un primer elemento sintáctico que especifica si un valor de contador del orden de imágenes (POC) de una imagen actual se restablece para que sea igual a cero, cuando el primer elemento sintáctico especifica que el valor de POC de la imagen actual se restablece para que sea igual a cero, si la imagen actual está en una capa mayor que la capa 0, restablecer la imagen actual para que sea igual a cero y codificar la imagen actual utilizando el valor de restablecimiento de POC. La imagen actual puede ser una imagen no IDR. El elemento sintáctico puede comprender un indicador, como un `idr_au_present_flag`, y el indicador puede señalizarse en una cabecera de fragmento. Cuando la imagen es una imagen IDR, el elemento sintáctico siempre especifica que el valor de POC de la imagen actual se restablece para que sea igual a cero.

30 **[0147]** La FIG. 2 es un diagrama de bloques que ilustra un codificador de vídeo 20 de ejemplo que puede implementar las técnicas descritas en esta divulgación. 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 de un vídeo dentro de una trama o imagen de vídeo dada. La intercodificación se basa en la predicción temporal para reducir o eliminar la redundancia temporal de un vídeo dentro de tramas o imágenes adyacentes de una secuencia de vídeo. El intramodo (modo I) se puede referir a cualquiera de varios modos de compresión espacial. Los intermodos, tales como la predicción unidireccional (modo P) o la bipredicción (modo B), pueden hacer referencia a cualquiera de varios modos de compresión temporal.

35 **[0148]** En el ejemplo de la FIG. 2, el codificador de vídeo 20 incluye una unidad de partición 40, una unidad de procesamiento de predicción 42, una memoria de imágenes de referencia 64, un sumador 50, una unidad de procesamiento de transformada 52, una unidad de procesamiento de cuantificación 54 y una unidad de codificación por entropía 56. La unidad de procesamiento de predicción 42 incluye una unidad de estimación de movimiento 44, una unidad de compensación de movimiento 46 y una unidad de intrapredicción 48. Para la reconstrucción de bloques de vídeo, el codificador de vídeo 20 también incluye una unidad de procesamiento de cuantificación inversa 58, una unidad de procesamiento de transformada inversa 60 y un sumador 62. También se puede incluir un filtro de eliminación de bloques (no se muestra en la FIG. 2) para filtrar los límites de bloque, para eliminar distorsiones de efecto pixelado del vídeo reconstruido. Si se desea, el filtro de eliminación de bloques filtrará típicamente la salida del sumador 62. También se pueden usar filtros de bucle adicionales (en el bucle o tras el bucle), además del filtro de eliminación de bloques.

50 **[0149]** En diversos ejemplos, una unidad de codificador de vídeo 20 puede estar encargada de realizar las técnicas de esta divulgación. También, en algunos ejemplos, las técnicas de esta divulgación pueden dividirse entre una o más de las unidades del codificador de vídeo 20.

55 **[0150]** Como se muestra en la FIG. 2, el codificador de vídeo 20 recibe datos de vídeo, y la unidad de partición 40 divide los datos en bloques de vídeo. Esta partición también puede incluir la partición en fragmentos, mosaicos u otras unidades mayores, así como la división en bloques de vídeo, por ejemplo, de acuerdo con una estructura de árbol cuaternario de LCU y CU. El codificador de vídeo 20 ilustra, en general, los componentes que codifican bloques de vídeo dentro de un fragmento de vídeo que se va a codificar. El fragmento se puede dividir en múltiples bloques de vídeo (y, posiblemente, en conjuntos de bloques de vídeo denominados mosaicos). La unidad de procesamiento de predicción 42 puede seleccionar uno entre una pluralidad de posibles modos de codificación, tal como uno entre una pluralidad de modos de intracodificación, o uno entre una pluralidad de modos de intercodificación, para el bloque de vídeo actual basándose en resultados de error (por ejemplo, la velocidad de codificación y el nivel de distorsión). La unidad de procesamiento de predicción 42 puede proporcionar el bloque intracodificado o intercodificado resultante al sumador 50 para generar datos de bloque residuales y al sumador 62 para reconstruir el bloque codificado para su uso como imagen de referencia.

65

[0151] La unidad de intrapredicción 48, dentro de la unidad de procesamiento de predicción 42, puede realizar la codificación intrapredictiva del bloque de vídeo actual con respecto a uno o más bloques vecinos en la misma imagen o fragmento que el bloque actual a codificar, para proporcionar compresión espacial. La unidad de estimación de movimiento 44 y la unidad de compensación de movimiento 46 dentro de la unidad de procesamiento de predicción 42 realizan la codificación interpredictiva del bloque de vídeo actual con respecto a uno o más bloques predictivos en una o más imágenes de referencia, para proporcionar compresión temporal.

[0152] La unidad de estimación de movimiento 44 se puede configurar para determinar el modo de interpredicción para un fragmento de vídeo de acuerdo con un patrón predeterminado para una secuencia de vídeo. La unidad de estimación de movimiento 44 y la unidad de compensación de movimiento 46 pueden estar altamente integradas, pero se ilustran por separado con propósitos conceptuales. La estimación del movimiento, realizada por la unidad de estimación de movimiento 44, es el proceso de generación de vectores de movimiento, que estiman el movimiento para los bloques de vídeo. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de una PU de un bloque de vídeo dentro de una trama o imagen de vídeo actual en relación con un bloque predictivo de una imagen de referencia.

[0153] Un bloque predictivo es un bloque que se determina que coincide estrechamente con la PU del bloque de vídeo que se va a codificar en términos de diferencia de píxel, que se puede determinar mediante una suma de diferencia absoluta (SAD), una suma de diferencia al cuadrado (SSD) u otras métricas de diferencia. En algunos ejemplos, el codificador de vídeo 20 puede calcular valores para posiciones de píxeles fraccionarios de las imágenes de referencia almacenadas en la memoria de imágenes de referencia 64. Por ejemplo, el codificador de vídeo 20 puede interpolar valores de posiciones de un cuarto de píxel, posiciones de un octavo de píxel u otras posiciones fraccionarias de píxel de la imagen de referencia. Por lo tanto, la unidad de estimación de movimiento 44 puede realizar una búsqueda de movimiento con respecto a las posiciones de píxel completo y las posiciones de píxel fraccionario, y proporcionar un vector de movimiento con una precisión de píxel fraccionario.

[0154] La unidad de estimación de movimiento 44 calcula un vector de movimiento para una PU de un bloque de vídeo en un fragmento intercodificado, comparando la posición de la PU con la posición de un bloque predictivo de una imagen de referencia. La imagen de referencia puede ser seleccionada a partir de una primera lista de imágenes de referencia (Lista 0) o de una segunda lista de imágenes de referencia (Lista 1), cada una de las cuales identifica una o más imágenes de referencia almacenadas en una memoria de imágenes de referencia 64. La unidad de estimación de movimiento 44 envía el vector de movimiento calculado a la unidad de codificación por entropía 56 y a la unidad de compensación de movimiento 46.

[0155] La compensación de movimiento, realizada por la unidad de compensación de movimiento 46, puede implicar extraer o generar el bloque predictivo en base al vector de movimiento determinado mediante estimación de movimiento, realizando posiblemente interpolaciones con una precisión de subpíxel. Tras recibir el vector de movimiento para la PU del bloque de vídeo actual, la unidad de compensación de movimiento 46 puede localizar el bloque predictivo al que apunta el vector de movimiento en una de las listas de imágenes de referencia. El codificador de vídeo 20 forma un bloque de vídeo residual restando los valores de píxel del bloque predictivo a los valores de píxel del bloque de vídeo actual que se está codificando, formando valores de diferencia de píxel. Los valores de diferencia de píxel forman datos residuales para el bloque, y pueden incluir componentes de diferencia tanto de luma como de croma. El sumador 50 representa el componente o los componentes que realizan esta operación de resta. La unidad de compensación de movimiento 46 también puede generar elementos sintácticos asociados a los bloques de vídeo y al fragmento de vídeo para su uso por el descodificador de vídeo 30 en la descodificación de los bloques de vídeo del fragmento de vídeo.

[0156] Además, en algunos casos, la unidad de procesamiento de predicción 42 puede determinar predecir un bloque de una imagen usando predicción entre capas. Por ejemplo, para los datos de vídeo multivista, la unidad de procesamiento de predicción 42 puede determinar predecir un bloque de una imagen de una vista a partir de una imagen de otra vista usando la predicción entre vistas. En el caso de la predicción entre vistas, la unidad de estimación de movimiento 44 puede calcular un vector de movimiento de disparidad para el bloque, donde el vector de movimiento de disparidad en general identifica la posición de un bloque de referencia en una imagen de una vista de referencia. Para mencionar otro ejemplo, para la codificación de vídeo escalable, la unidad de procesamiento de predicción 42 puede determinar predecir un bloque de una imagen de una capa a partir de una imagen de otra capa usando predicción entre capas.

[0157] La predicción entre capas (que incluye la predicción entre vistas, ya que las vistas pueden considerarse un tipo de capa) puede implicar el uso de vectores de movimiento que apuntan a bloques de referencia en diferentes capas, en lugar de bloques en la misma capa pero en imágenes de diferentes instancias temporales. Típicamente, la predicción entre capas se realiza utilizando imágenes de la misma unidad de acceso que una imagen que incluye un bloque a predecir utilizando la predicción entre capas. Los parámetros de movimiento para un bloque interpredictivo pueden por lo tanto identificar una imagen de referencia utilizando, por ejemplo, un valor de contador del orden de imágenes (POC).

[0158] Típicamente, las técnicas de las extensiones de codificación entre capas (por ejemplo, extensiones multivista y escalables a la HEVC) se han basado en el supuesto de que los valores de POC están alineados, es decir, que todas las imágenes en la misma unidad de acceso tienen el mismo valor de POC. Para lograr esto, las técnicas convencionales también han alineado imágenes con punto de acceso intraaleatorio (IRAP). Sin embargo, como se ha señalado anteriormente, forzar la alineación de las imágenes IRAP dificulta ciertos escenarios de utilización ventajosos.

[0159] Por lo tanto, de acuerdo con las técnicas de esta divulgación, la unidad de procesamiento de predicción 42 no alinea necesariamente todas las imágenes IRAP entre capas diferentes. Es decir, la unidad de procesamiento de predicción 42 puede determinar codificar al menos una imagen de una unidad de acceso que incluye una imagen IRAP como una imagen no IRAP. Sin embargo, la unidad de procesamiento de predicción 42 puede usar las técnicas de esta divulgación para lograr imágenes IRAP sin alineación al mismo tiempo que se mantienen los valores de POC alineados entre las imágenes de capas diferentes. De esta manera, la unidad de procesamiento de predicción 42 puede asegurar que los bloques de imágenes predictivos entre capas puedan referir apropiadamente a imágenes de referencia en capas diferentes donde las imágenes de referencia tienen los mismos valores de POC que las imágenes que incluyen los bloques predictivos entre capas, sin alinear imágenes IRAP.

[0160] En particular, de acuerdo con las técnicas de esta divulgación, la unidad de procesamiento de predicción 42 puede recibir una imagen a codificar para una unidad de acceso que incluye una imagen IRAP. Es decir, el codificador de vídeo 20 puede haber codificado previamente una capa base (u otra capa) que incluye la imagen IRAP, y puede haber determinado codificar la imagen IRAP como una imagen IRAP, por ejemplo, una imagen IDR, CRA o BLA. Sin embargo, el codificador de vídeo 20 puede determinar codificar una imagen coubicada (es decir, una imagen en la misma unidad de acceso) en otra capa como una imagen no IRAP.

[0161] Después de determinar codificar una imagen como una imagen no IRAP que está en la misma unidad de acceso que una imagen IRAP, el codificador de vídeo 20 puede establecer un valor para un elemento sintáctico que indica que al menos una porción de un valor de POC para la imagen IRAP se debe restablecer en un valor de cero. La al menos porción del valor de POC puede incluir solamente los bits más significativos (MSB) del valor de POC, o el valor de POC completo (es decir, todos los bits del valor de POC). El elemento sintáctico puede comprender, por ejemplo, el indicador de restablecimiento de los MSB de POC del Ejemplo 1 anterior o el indicador de restablecimiento de POC del Ejemplo 2 anterior.

[0162] El codificador de vídeo 20 también puede restablecer el valor de POC para la imagen después de determinar codificar la imagen como una imagen no IRAP. Además, el codificador de vídeo 20 puede restablecer los valores de POC de las imágenes en la memoria de imágenes de referencia 64 que están en la misma capa que la imagen y en la misma secuencia de vídeo codificada que la imagen. En particular, restablecer los valores de POC de estas imágenes puede incluir asegurar que las diferencias entre el valor de POC para la imagen actual y los valores de POC para estas otras imágenes permanezcan iguales antes y después del restablecimiento. Por lo tanto, si una imagen en la memoria de imágenes de referencia 64 tiene una diferencia de valor de POC en relación con el valor de POC de la imagen actual de N, el codificador de vídeo 20 puede restablecer el valor de POC de la imagen en la memoria de imágenes de referencia 64 de manera que el valor de POC sea igual para restablecer el valor de POC de la imagen actual menos N.

[0163] De esta manera, al codificar imágenes posteriores (por ejemplo, imágenes de la misma capa o capas codificadas posteriormente) usando la predicción entre capas, el codificador de vídeo 20 puede codificar datos sintácticos que identifican una imagen de referencia usando un valor de POC para la imagen de referencia que es igual a un valor de POC para una imagen que incluye un bloque predictivo entre capas que se refiere a la imagen de referencia.

[0164] La unidad de intrapredicción 48 puede intrapredicir un bloque actual, como alternativa a la interpredicción llevada a cabo por la unidad de estimación de movimiento 44 y la unidad de compensación de movimiento 46, como se describe anteriormente. En particular, la unidad de intrapredicción 48 puede determinar un modo de intrapredicción a usar para codificar un bloque actual. En algunos ejemplos, la unidad de intrapredicción 48 puede codificar un bloque actual usando diversos modos de intrapredicción, por ejemplo, durante pasadas de codificación independientes, y la unidad de intrapredicción 48 puede seleccionar un modo de intrapredicción adecuado a usar partir de los modos sometidos a prueba. Por ejemplo, la unidad de intrapredicción 48 puede calcular valores de velocidad-distorsión usando un análisis de velocidad-distorsión para los diversos modos de intrapredicción sometidos a prueba, y seleccionar el modo de intrapredicción que tenga las mejores características de velocidad-distorsión entre los modos sometidos a prueba. El análisis de velocidad-distorsión determina, en general, una cantidad de distorsión (o error) entre un bloque codificado y un bloque original no codificado que se codificó para producir el bloque codificado, así como una velocidad de transmisión de bits (es decir, un número de bits) usada para producir el bloque codificado. La unidad de intrapredicción 48 puede calcular proporciones a partir de las distorsiones y velocidades de los diversos bloques codificados, para determinar qué modo de intrapredicción presenta el mejor valor de velocidad-distorsión para el bloque.

[0165] En cualquier caso, tras seleccionar un modo de intrapredicción para un bloque, la unidad de intrapredicción 48 puede proporcionar a la unidad de codificación por entropía 56 información que indica el modo de intrapredicción seleccionado para el bloque. La unidad de codificación por entropía 56 puede codificar la información que indica el modo de intrapredicción seleccionado de acuerdo con las técnicas de esta divulgación. El codificador de vídeo 20 puede incluir, en el flujo de bits transmitido, datos de configuración, que pueden incluir una pluralidad de tablas de índices del modo de intrapredicción y una pluralidad de tablas de índices del modo de intrapredicción modificadas (también denominadas tablas de correlación de palabras de código), definiciones de contextos de codificación para diversos bloques e indicaciones de un modo de intrapredicción más probable, una tabla de índices del modo de intrapredicción y una tabla de índices del modo de intrapredicción modificada, a usar para cada uno de los contextos.

[0166] Después de que la unidad de procesamiento de predicción 42 genere el bloque predictivo para el bloque de vídeo actual, ya sea mediante interpredicción o intrapredicción, el codificador de vídeo 20 forma un bloque de vídeo residual restando el bloque predictivo al bloque de vídeo actual. Los datos de vídeo residuales del bloque residual se pueden incluir en una o más TU y aplicarse a la unidad de procesamiento de transformada 52. La unidad de procesamiento de transformada 52 transforma los datos de vídeo residuales en coeficientes de transformada residuales usando una transformada, tal como una transformada discreta del coseno (DCT) o una transformada conceptualmente similar. La unidad de procesamiento de transformada 52 puede convertir los datos de vídeo residuales de un dominio de píxel a un dominio de transformada, tal como un dominio de la frecuencia.

[0167] La unidad de procesamiento de transformada 52 puede enviar los coeficientes de transformada resultantes a la unidad de procesamiento de cuantificación 54. La unidad de procesamiento de cuantificación 54 cuantifica los coeficientes de transformada para reducir aún más la velocidad de transmisión de bits. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos, o a la totalidad, de los coeficientes. El grado de cuantificación se puede modificar ajustando un parámetro de cuantificación. En algunos ejemplos, la unidad de procesamiento de cuantificación 54 luego puede realizar una exploración de la matriz, incluyendo los coeficientes de transformada cuantificados. De forma alternativa, la unidad de codificación por entropía 56 puede realizar la exploración.

[0168] Tras la cuantificación, la unidad de codificación por entropía 56 codifica por entropía los coeficientes de transformada cuantificados. Por ejemplo, la unidad de codificación por entropía 56 puede realizar una codificación de longitud variable adaptativa al contexto (CAVLC), una codificación aritmética binaria adaptativa al contexto (CABAC), una codificación aritmética binaria adaptativa al contexto basada en la sintaxis (SBAC), una codificación por entropía por división de intervalos de probabilidad (PIPE) u otra metodología o técnicas de codificación por entropía. Tras la codificación por entropía mediante la unidad de codificación por entropía 56, el flujo de bits codificado se puede transmitir al descodificador de vídeo 30, o archivarse para su posterior transmisión o recuperación mediante el descodificador de vídeo 30. La unidad de codificación por entropía 56 también puede codificar por entropía los vectores de movimiento y los otros elementos sintácticos para el fragmento de vídeo actual que se está codificando.

[0169] La unidad de procesamiento de cuantificación inversa 58 y la unidad de procesamiento de transformada inversa 60 aplican una cuantificación inversa y una transformada inversa, respectivamente, para reconstruir el bloque residual en el dominio de píxel, para su posterior uso como un bloque de referencia de una imagen de referencia. La unidad de compensación de movimiento 46 puede calcular un bloque de referencia añadiendo el bloque residual a un bloque predictivo de una de las imágenes de referencia de una de las listas de imágenes de referencia. La unidad de compensación de movimiento 46 también puede aplicar uno o más filtros de interpolación al bloque residual reconstruido para calcular valores de píxeles fraccionarios para su uso en la estimación de movimiento. El sumador 62 añade el bloque residual reconstruido al bloque de predicción compensado por movimiento, generado por la unidad de compensación de movimiento 46 para producir un bloque de referencia para su almacenamiento en la memoria de imágenes de referencia 64. La unidad de estimación de movimiento 44 y la unidad de compensación de movimiento 46 pueden usar el bloque de referencia como bloque de referencia para intrapredicir un bloque en una imagen o trama de vídeo posterior.

[0170] De esta manera, el codificador de vídeo 20 representa un ejemplo de un codificador de vídeo configurado para codificar un valor para un elemento sintáctico que indica si al menos una porción del valor de contador del orden de imágenes (POC) de una imagen se debe restablecer en un valor de cero, cuando el valor del elemento sintáctico indica que la porción del valor de POC se debe restablecer en el valor de cero, restablecer al menos la porción del valor de POC de manera que la porción del valor de POC sea igual a cero, y codificar los datos de vídeo utilizando el valor de restablecimiento de POC.

[0171] La FIG. 3 es un diagrama de bloques que ilustra un descodificador de vídeo 30 de ejemplo que puede implementar las técnicas descritas en esta divulgación. En el ejemplo de la FIG. 3, el descodificador de vídeo 30 incluye una unidad de descodificación por entropía 70, una unidad de procesamiento de predicción 71, una unidad de procesamiento de cuantificación inversa 76, una unidad de transformada inversa 78, un sumador 80 y una memoria de imágenes de referencia 82. La unidad de procesamiento de predicción 71 incluye una unidad de compensación de movimiento 72 y una unidad de intrapredicción 74. En algunos ejemplos, el descodificador de vídeo 30 realiza una pasada de descodificación que, en general, es recíproca a la pasada de codificación descrita con respecto al codificador de vídeo 20 de la FIG. 2.

[0172] En diversos ejemplos, una unidad de descodificador de vídeo 30 puede estar encargada de realizar las técnicas de esta divulgación. También, en algunos ejemplos, las técnicas de esta divulgación pueden dividirse entre una o más de las unidades del descodificador de vídeo 30.

5 **[0173]** Durante el procedimiento de descodificación, el descodificador de vídeo 30 recibe un flujo de bits de vídeo codificado, que representa bloques de vídeo de un fragmento de vídeo codificado y elementos sintácticos asociados, desde el codificador de vídeo 20. La unidad de descodificación por entropía 70 del descodificador de vídeo 30 descodifica por entropía el flujo de bits para generar coeficientes cuantificados, vectores de movimiento y otros elementos sintácticos. La unidad de descodificación por entropía 70 reenvía los vectores de movimiento y otros
10 elementos sintácticos a la unidad de procesamiento de predicción 71. El descodificador de vídeo 30 puede recibir los elementos sintácticos a nivel de fragmento de vídeo y/o a nivel de bloque de vídeo.

[0174] Cuando el fragmento de vídeo se codifica como un fragmento intracodificado (I), la unidad de intrapredicción 74 de la unidad de procesamiento de predicción 71 puede generar datos de predicción para un bloque de vídeo del fragmento de vídeo actual, basándose en un modo de intrapredicción señalado y en datos de bloques previamente descodificados de la trama o imagen actual. Cuando la trama de vídeo se codifica como un fragmento intercodificado (es decir, B o P), la unidad de compensación de movimiento 72 de la unidad de procesamiento de predicción 71 produce bloques predictivos para un bloque de vídeo del fragmento de vídeo actual, basándose en los vectores de movimiento y otros elementos sintácticos recibidos desde la unidad de descodificación por entropía 70. Los bloques predictivos se pueden generar a partir de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. El descodificador de vídeo 30 puede construir las listas de imágenes de referencia, la Lista 0 y la Lista 1, usando técnicas de construcción por defecto basándose en las imágenes de referencia almacenadas en la memoria de imágenes de referencia 82.

25 **[0175]** La unidad de compensación de movimiento 72 determina la información de predicción para un bloque de vídeo del fragmento de vídeo actual, analizando sintácticamente los vectores de movimiento y otros elementos sintácticos, y usa la información de predicción para producir los bloques predictivos para el bloque de vídeo actual que se está descodificando. Por ejemplo, la unidad de compensación de movimiento 72 usa algunos de los elementos sintácticos recibidos para determinar un modo de predicción (por ejemplo, intrapredicción o interpredicción), usado
30 para codificar los bloques de vídeo del fragmento de vídeo, un tipo de fragmento de interpredicción (por ejemplo, fragmento B o fragmento P), información de construcción para una o más de las listas de imágenes de referencia para el fragmento, vectores de movimiento para cada bloque de vídeo intercodificado del fragmento, el estado de interpredicción para cada bloque de vídeo intercodificado del fragmento y otra información para descodificar los bloques de vídeo en el fragmento de vídeo actual.

35 **[0176]** La unidad de compensación de movimiento 72 también puede realizar la interpolación basándose en filtros de interpolación. La unidad de compensación de movimiento 72 puede usar filtros de interpolación como los usados por el codificador de vídeo 20 durante la codificación de los bloques de vídeo para calcular valores interpolados de píxeles fraccionarios de los bloques de referencia. En este caso, la unidad de compensación de movimiento 72 puede
40 determinar los filtros de interpolación usados por el codificador de vídeo 20 a partir de los elementos sintácticos recibidos y usar los filtros de interpolación para producir bloques predictivos.

[0177] De acuerdo con las técnicas de esta divulgación, la unidad de descodificación por entropía 70 puede descodificar un valor para un elemento sintáctico que indica si al menos una porción del valor de un primer contador del orden de imágenes (POC) de una imagen se debe restablecer en un valor de cero. El elemento sintáctico puede comprender, por ejemplo, poc_msb_reset_flag del Ejemplo 1 como se describe anteriormente o el indicador de restablecimiento de poc del Ejemplo 2 como se describe anteriormente. El descodificador de vídeo 30 puede restablecer la al menos porción del valor de POC cuando el elemento sintáctico tiene un valor que indica que el valor de POC se va a restablecer. Por ejemplo, el descodificador de vídeo 30 puede establecer bits para la al menos porción
50 (por ejemplo, los bits más significativos para el indicador de restablecimiento de los MSB de POC o todos los bits para el poc_reset_flag) del valor de POC igual a cero.

[0178] Además, el descodificador de vídeo 30 también puede ajustar los valores de POC de otras imágenes en la memoria de imágenes de referencia 82 (es decir, aquellas imágenes que están en la misma capa de codificación de vídeo y la misma secuencia de vídeo codificada). Por ejemplo, el descodificador de vídeo 30 puede reducir los valores de POC de las otras imágenes de manera que las diferencias entre los valores de POC y el valor de POC para el que se señaló el restablecimiento permanecen iguales que antes del restablecimiento. Por lo tanto, si una imagen en la memoria de imágenes de referencia 82 tiene una diferencia de valor de POC en relación con el valor de POC de la imagen actual de N, el descodificador de vídeo 30 puede restablecer el valor de POC de la imagen en la memoria de imágenes de referencia 82 de modo que el valor de POC sea igual para restablecer el valor de POC de la imagen actual menos N.

[0179] En general, los bloques de vídeo interpredictivos pueden incluir elementos sintácticos que identifican bloques de referencia, ya sea que la predicción sea temporal o entre capas (por ejemplo, entre vistas). De acuerdo con las técnicas de esta divulgación, los elementos sintácticos para los bloques interpredictivos pueden identificar una imagen de referencia usando un valor de POC de la imagen de referencia después de un restablecimiento como se ha

analizado anteriormente. Por lo tanto, el descodificador de vídeo 30 no necesita ajustar los valores de POC que identifican las imágenes de referencia cuando se señalizan como información de movimiento con los fines de interpredicción.

5 **[0180]** La unidad de procesamiento de cuantificación inversa 76 cuantifica de manera inversa, es decir, descuantifica, los coeficientes de transformada cuantificados, proporcionados en el flujo de bits y descodificados por la unidad de descodificación por entropía 70. El proceso de cuantificación inversa puede incluir el uso de un parámetro de cuantificación calculado por el codificador de vídeo 20 para cada bloque de vídeo del fragmento de vídeo para determinar un grado de cuantificación y, asimismo, un grado de cuantificación inversa que se debería aplicar. La
10 unidad de procesamiento de transformada inversa 78 aplica una transformada inversa, por ejemplo una DCT inversa, una transformada entera inversa o un proceso de transformada inversa conceptualmente similar, a los coeficientes de transformada con el fin de producir bloques residuales en el dominio de píxel.

15 **[0181]** Una vez que la unidad de compensación de movimiento 72 genera el bloque predictivo para el bloque de vídeo actual basándose en los vectores de movimiento y en otros elementos sintácticos, el descodificador de vídeo 30 forma un bloque de vídeo descodificado sumando los bloques residuales procedentes de la unidad de procesamiento de transformada inversa 78 a los bloques predictivos correspondientes generados por la unidad de compensación de movimiento 72. El sumador 80 representa el componente o los componentes que realizan esta operación de suma. Si se desea, también se puede aplicar un filtro de eliminación de bloques para filtrar los bloques
20 decodificados a fin de eliminar distorsiones de efecto pixelado. También se pueden usar otros filtros de bucle (bien en el bucle de codificación o bien después del bucle de codificación) para suavizar las transiciones de píxeles o mejorar de otro modo la calidad del vídeo. Los bloques de vídeo descodificados en una trama o imagen dada se almacenan a continuación en la memoria de imágenes de referencia 82, que almacena imágenes de referencia usadas para una compensación de movimiento posterior. La memoria de imágenes de referencia 82 también almacena vídeo
25 descodificado para su presentación posterior en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la FIG. 1.

30 **[0182]** En algunos ejemplos, uno o más aspectos de las técnicas descritas en esta divulgación pueden ser realizados por un dispositivo de red intermedio, tal como un elemento de red sensible a los medios (MANE), un procesador de adaptación de flujo, un procesador de empalme o un procesador de edición. Por ejemplo, dicho dispositivo intermedio puede configurarse para generar cualquiera de una variedad de señalización como se describe en esta divulgación. Por ejemplo, dicho dispositivo intermedio se puede configurar para recibir un primer elemento sintáctico que especifica si un valor de contador del orden de imágenes (POC) de una imagen actual se restablece para que sea igual a un conjunto de los bits menos significativos (LSB) del valor de POC de la imagen actual, cuando el primer elemento
35 sintáctico especifica que el valor de POC de la imagen actual se restablece para que sea igual al conjunto de LSB del valor de POC de la imagen actual, disminuya los valores de POC de otras imágenes en una misma capa de codificación como la imagen actual de modo que la diferencia entre el valor de POC de la imagen actual y el valor de POC de cualquiera de las imágenes en la misma capa permanezca igual que antes de que se restableciera el valor de POC de la imagen actual, y descodifique al menos parte de imagen actual y las otras imágenes usando los valores de POC disminuidos.

40 **[0183]** En otro ejemplo, dicho dispositivo intermedio se puede configurar para recibir un primer elemento sintáctico que especifica si un valor de contador del orden de imágenes (POC) de una imagen actual se restablece para que sea igual a cero, cuando el primer elemento sintáctico especifica que el valor de POC de la imagen actual se restablece para que sea igual a cero, si la imagen actual está en una capa mayor que la capa 0, restablecer la imagen actual para que sea igual a cero y descodificar la imagen actual utilizando el valor de restablecimiento de POC.

45 **[0184]** De esta manera, el descodificador de vídeo 30 representa un ejemplo de un codificador de vídeo configurado para codificar un valor para un elemento sintáctico que indica si al menos una porción del valor de contador del orden de imágenes (POC) de una imagen se debe restablecer en un valor de cero, cuando el valor del elemento sintáctico indica que la porción del valor de POC se debe restablecer en el valor de cero, restablecer al menos la porción del valor de POC de manera que la porción del valor de POC sea igual a cero, y codificar los datos de vídeo utilizando el valor de restablecimiento de POC.

50 **[0185]** La FIG. 4 es un diagrama conceptual que ilustra una secuencia de imágenes de vídeo codificadas 100-132. Las imágenes están sombreadas de manera diferente para indicar las posiciones dentro de una estructura de predicción jerárquica. Por ejemplo, las imágenes 100, 116 y 132 están sombreadas en negro para representar que las imágenes 100, 116, 132 están en la parte superior de la estructura de predicción jerárquica. Las imágenes 100, 116, 132 pueden comprender, por ejemplo, imágenes intracodificadas o imágenes intercodificadas que se predicen a partir
55 de otras imágenes en una única dirección (por ejemplo, imágenes P). Cuando están intracodificadas, las imágenes 100, 116, 132 se predicen únicamente a partir de datos dentro de la misma imagen. Cuando están intercodificadas, la imagen 116, por ejemplo, puede codificarse en relación con los datos de la imagen 100, como se indica con la flecha discontinua desde la imagen 116 a la imagen 100. Las imágenes 116, 132 forman imágenes clave de los grupos de imágenes (GOP) 134, 136, respectivamente.

65

[0186] Las imágenes 108, 124 están oscurecidas para indicar que se encuentran a continuación en la jerarquía de codificación después de las imágenes 100, 116 y 132. Las imágenes 108, 124 pueden comprender imágenes bidireccionales codificadas en modo de interpredicción. Por ejemplo, la imagen 108 puede predecirse a partir de los datos de las imágenes 100 y 116, mientras que la imagen 124 puede predecirse a partir de las imágenes 116 y 132. Las imágenes 104, 112, 120 y 128 están ligeramente sombreadas para indicar que se encuentran continuación en la jerarquía de codificación después de las imágenes 108 y 124. Las imágenes 104, 112, 120 y 128 también pueden comprender imágenes bidireccionales codificadas en modo de interpredicción. Por ejemplo, la imagen 104 puede predecirse a partir de las imágenes 100 y 108, la imagen 112 puede predecirse a partir de las imágenes 108 y 116, la imagen 120 puede predecirse a partir de las imágenes 116 y 124, y la imagen 128 puede predecirse a partir de la imagen 124 y 132.

[0187] Finalmente, las imágenes 102, 106, 110, 114, 118, 122, 126 y 130 son sombreadas de color blanco para indicar que estas imágenes son las últimas en la jerarquía de codificación. Las imágenes 102, 106, 110, 114, 118, 122, 126 y 130 pueden ser imágenes bidireccionales codificadas en modo de interpredicción. La imagen 102 puede predecirse a partir de las imágenes 100 y 104, la imagen 106 puede predecirse a partir de las imágenes 104 y 108, la imagen 110 puede predecirse a partir de las imágenes 108 y 112, la imagen 114 puede predecirse a partir de las imágenes 112 y 116, la imagen 118 puede predecirse a partir de las imágenes 116 y 120, la imagen 122 puede predecirse a partir de las imágenes 120 y 124, la imagen 126 puede predecirse a partir de las imágenes 124 y 128, y la imagen 130 puede predecirse a partir de las imágenes 128 y 132.

[0188] Las imágenes 100-132 se ilustran en orden de visualización. Es decir, después de la descodificación, la imagen 100 se visualiza antes que la imagen 102, la imagen 102 se visualiza antes que la imagen 104, y así sucesivamente. Sin embargo, debido a la jerarquía de codificación, las imágenes 100-132 pueden descodificarse en un orden diferente. Además, mientras son codificadas, las imágenes 100-132 pueden disponerse en el orden de descodificación en un flujo de bits que incluye datos codificados para las imágenes 100-132. Por ejemplo, la imagen 116 se puede visualizar la última entre las imágenes del GOP 134. Sin embargo, debido a la jerarquía de codificación, la imagen 116 puede descodificarse como la primera del GOP 134. Es decir, a fin de descodificar apropiadamente la imagen 108, por ejemplo, la imagen 116 puede necesitar ser descodificada primero, a fin de actuar como una imagen de referencia para la imagen 108. Asimismo, la imagen 108 puede actuar como una imagen de referencia para las imágenes 104, 106, 110 y 112, y por lo tanto puede necesitar ser descodificada antes que las imágenes 104, 106, 110 y 112.

[0189] El tiempo en el que se visualiza una imagen puede denominarse tiempo de presentación, mientras que el tiempo en el que se descodifica la imagen puede denominarse tiempo de descodificación. Los tiempos de descodificación y presentación en general proporcionan indicaciones de ordenamiento temporal en relación con otras imágenes de la misma secuencia. La diferencia entre el tiempo de descodificación de una imagen y el tiempo de presentación de la imagen puede denominarse retardo en el reordenamiento de la imagen.

[0190] De acuerdo con las técnicas de esta divulgación, un codificador de vídeo (por ejemplo, codificador de vídeo 20 y/o descodificador de vídeo 30) puede ajustar los valores de POC para las imágenes en una capa de codificación de vídeo cuando una de las imágenes es una imagen no IRAP incluida en una unidad de acceso que también incluye una imagen IRAP (de una capa de codificación de vídeo diferente). Por ejemplo, suponga que la imagen 132 es una imagen P y está ubicada con (es decir, incluida en la misma unidad de acceso que) una imagen de otra capa de codificación de vídeo que es una imagen IRAP. En este ejemplo, la imagen 132 sería una imagen no IRAP (porque está intercodificada) que se incluye en la misma unidad de acceso que una imagen IRAP.

[0191] Por consiguiente, siguiendo con el ejemplo expuesto anteriormente, el codificador de vídeo codificaría un valor para un elemento sintáctico que indica que un valor de POC para la imagen 132 se debe restablecer. Es decir, el elemento sintáctico indicaría que al menos una porción del valor de POC para la imagen 132 se debe restablecer (en un valor de cero). Por consiguiente, el codificador de vídeo puede ajustar el valor de POC de la imagen 132. Por ejemplo, de acuerdo con el Ejemplo 1 analizado anteriormente en el que el elemento sintáctico es `poc_msb_reset_flag`, el codificador de vídeo establecería los bits más significativos (MSB) del valor de POC de la imagen 132 igual a cero. Para mencionar otro ejemplo, de acuerdo con el Ejemplo 2 analizado anteriormente en el que el elemento sintáctico es `poc_reset_flag`, el codificador de vídeo establecería el valor de POC completo de la imagen 132 igual a cero.

[0192] El codificador de vídeo también puede ajustar los valores de POC de otras imágenes. Por ejemplo, suponiendo que las imágenes 100 y 116 fueron codificadas previamente, el codificador de vídeo puede ajustar los valores de POC de las imágenes 100 y 116 basándose en el valor de restablecimiento de POC para la imagen 132. En el ejemplo de la FIG. 4, el valor de POC de la imagen 132 es ocho veces mayor que el valor de POC de la imagen 116 y dieciséis veces mayor que el valor de POC de la imagen 100. En el Ejemplo 2, donde se restablece todo el valor de POC, el codificador de vídeo puede ajustar el valor de POC de la imagen 100 para que sea igual a -16 (negativo dieciséis), y el valor de POC de la imagen 116 para que sea igual a -8 (negativo ocho). De esta manera, las diferencias entre los valores de POC de la imagen 132 y las imágenes 100 y 116 pueden permanecer constantes después del restablecimiento del valor de POC, en relación con antes del restablecimiento del valor de POC.

[0193] El ejemplo anterior explica la modificación de los valores de POC de las imágenes 100 y 116 basándose en el supuesto de que estas imágenes ya estarían presentes en un búfer de imágenes descodificadas (DPB) al codificar la imagen 132. En algunos casos, las imágenes 102-114 también pueden estar presentes en el DPB, en cuyo caso el codificador de vídeo también puede igualmente ajustar los valores de POC de las imágenes 102-114. Sin embargo, debido a que las imágenes 118-130 pueden depender de la imagen 132, se espera que las imágenes 118-130 no estén presentes en el DPB cuando se restablezca la imagen 132. Es decir, las imágenes 118-130 tienen un orden de descodificación que es posterior al orden de descodificación de la imagen 132. Por lo tanto, el codificador de vídeo no necesita ajustar los valores de POC para las imágenes 118-130. En algunos ejemplos, las imágenes 102-114 también pueden tener un orden de descodificación que sea posterior al orden de descodificación de la imagen 132.

[0194] La FIG. 5 es un diagrama de flujo que ilustra un procedimiento de codificación de datos de vídeo de ejemplo de acuerdo con las técnicas de esta divulgación. El codificador de vídeo 20 se describe como que realiza el procedimiento de la FIG. 5. Sin embargo, se debe entender que otros dispositivos de codificación se pueden configurar para realizar este o un procedimiento similar. Además, las etapas del procedimiento pueden realizarse en un orden diferente, o en paralelo, y ciertas etapas pueden añadirse u omitirse.

[0195] En el ejemplo de la FIG. 5, el codificador de vídeo 20 codifica inicialmente imágenes de una primera capa de codificación de vídeo (150). Como se ha analizado anteriormente, la capa puede corresponder a una capa de codificación de vídeo escalable o una vista para la codificación de vídeo de multivista. El uso del término "primero" en este contexto pretende ser nominal en lugar de ordinal; en algunos ejemplos, el codificador de vídeo 20 puede haber codificado una o más capas adicionales de codificación de vídeo antes de la primera capa analizada con respecto a la FIG. 5. La codificación de imágenes de la primera capa de codificación de vídeo en general implica la codificación de ciertas imágenes de la primera capa como imágenes con punto de acceso intraaleatorio (IRAP) y otras imágenes como imágenes no IRAP, por ejemplo, imágenes interpredictivas (ya sean temporales o entre capas).

[0196] El codificador de vídeo 20 puede entonces determinar una imagen de una segunda capa de codificación de vídeo que está coubicada con una imagen IRAP de la primera capa de codificación de vídeo (152). Sin embargo, el codificador de vídeo 20 puede determinar codificar esta imagen de la segunda capa de codificación de vídeo como una imagen no IRAP (154). Por ejemplo, el codificador de vídeo 20 puede determinar codificar la imagen usando predicción temporal y/o predicción entre capas (por ejemplo, entre vistas).

[0197] Como resultado de que la imagen es una imagen no IRAP pero coubicada con una imagen IRAP (por ejemplo, incluida en la misma unidad de acceso que la imagen IRAP), el codificador de vídeo 20 puede codificar un valor para un elemento sintáctico que indica que al menos una porción de un valor de POC para la imagen se debe restablecer (156). Por ejemplo, el elemento sintáctico puede comprender el `poc_msb_reset_flag` del Ejemplo 1 analizado anteriormente, o el `poc_reset_flag` del Ejemplo 2 analizado anteriormente. Además, el codificador de vídeo 20 puede ajustar los valores de POC de la imagen y las imágenes codificadas previamente que están en la misma capa y en la misma secuencia de vídeo codificada (158). En particular, el codificador de vídeo 20 puede ajustar los valores de POC de las otras imágenes de manera que las diferencias entre los valores de POC de las otras imágenes y la imagen actual permanezcan iguales después de restablecer el valor de POC para la imagen actual como antes del restablecimiento.

[0198] El codificador de vídeo 20 puede codificar además la imagen (160). Es decir, para cada bloque de la imagen, el codificador de vídeo 20 puede determinar si debe intraprededir o interprededir (usando predicción temporal o entre capas) el bloque. El codificador de vídeo 20 puede descodificar la imagen (162) y almacenar la imagen descodificada en un búfer de imágenes descodificadas (DPB), por ejemplo, la memoria de imágenes de referencia 64. El codificador de vídeo 20 puede entonces codificar una porción de una imagen posterior (por ejemplo, una imagen en la misma capa que tiene un orden de codificación posterior o una imagen en la misma unidad de acceso de una capa diferente) en relación con la imagen descodificada (166). A fin de identificar la imagen descodificada, el codificador de vídeo 20 puede codificar el valor de restablecimiento de POC para la imagen descodificada como información de movimiento de un bloque de la imagen posterior (168).

[0199] De esta manera, el procedimiento de la FIG. 5 representa un ejemplo de un procedimiento que incluye determinar si se restablece al menos una porción de un valor de contador del orden de imágenes (POC) de una imagen a un valor de cero, en respuesta a la determinación de restablecer la al menos porción del valor de POC, restablecer al menos la porción del valor de POC de manera que la porción del primer valor de POC sea igual a cero y codificar un valor para un elemento sintáctico que indique que al menos la porción del valor de POC se debe restablecer en el valor de cero, y codificar datos de vídeo usando el valor de restablecimiento de POC.

[0200] La FIG. 6 es un diagrama de flujo que ilustra un procedimiento de descodificación de datos de vídeo de ejemplo de acuerdo con las técnicas de esta divulgación. El descodificador de vídeo 30 se describe como que realiza el procedimiento de la FIG. 6. Sin embargo, se debe entender que otros dispositivos de codificación se pueden configurar para realizar este o un procedimiento similar. Además, las etapas del procedimiento pueden realizarse en un orden diferente, o en paralelo, y ciertas etapas pueden añadirse u omitirse.

[0201] En el ejemplo de la FIG. 6, el descodificador de vídeo 30 descodifica inicialmente imágenes de una primera capa de codificación de vídeo (180). Como se ha analizado anteriormente, la capa puede corresponder a una capa de codificación de vídeo escalable o una vista para la codificación de vídeo de multivista. El uso del término “primero” en este contexto pretende ser nominal en lugar de ordinal; en algunos ejemplos, el descodificador de vídeo 30 puede haber descodificado una o más capas adicionales de codificación de vídeo antes de la primera capa analizada con respecto a la FIG. 6. La descodificación de imágenes de la primera capa de codificación de vídeo en general implica la descodificación de ciertas imágenes de la primera capa como imágenes con punto de acceso intraaleatorio (IRAP) y otras imágenes como imágenes no IRAP, por ejemplo, imágenes interpredictivas (ya sean temporales o entre capas).

[0202] El descodificador de vídeo 30 puede descodificar un elemento sintáctico para una imagen de una segunda capa de codificación de vídeo que está coubicada con (es decir, en la misma unidad de acceso que) una imagen IRAP de la primera capa de codificación de vídeo que indica que al menos una porción del valor de POC para la imagen se va a restablecer (182). Por ejemplo, el elemento sintáctico puede comprender el `msb_reset_flag` de POC del Ejemplo 1 analizado anteriormente, o el `poc_reset_flag` del Ejemplo 2 analizado anteriormente. Basado en el valor de este elemento sintáctico, el descodificador de vídeo 30 puede ajustar los valores de POC de la imagen y las imágenes descodificadas previamente que están en la misma capa y en la misma secuencia de vídeo codificada (184). En particular, el descodificador de vídeo 30 puede ajustar los valores de POC de las otras imágenes de manera que las diferencias entre los valores de POC de las otras imágenes y la imagen actual permanezcan iguales después de restablecer el valor de POC para la imagen actual como antes del restablecimiento.

[0203] El descodificador de vídeo 30 también puede descodificar la imagen (186) y almacenar la imagen descodificada en un búfer de imágenes descodificadas (DPB), por ejemplo, la memoria de imágenes de referencia 82 (188). La descodificación de la imagen puede incluir bloques de descodificación de la imagen utilizando intrapredicción, interpredicción temporal y/o predicción entre capas. Cuando se realiza la predicción entre capas, el descodificador de vídeo 30 puede descodificar un valor de POC para una imagen de referencia en otra capa (por ejemplo, la imagen IRAP en la primera capa) que es igual al valor de restablecimiento de POC para la imagen actual. Al realizar una predicción temporal, el descodificador de vídeo 30 puede descodificar un valor de POC para una imagen de referencia en el DPB, donde el valor de POC descodificado puede corresponder a uno de los valores de POC ajustados para una de las imágenes descodificadas previamente en el DPB.

[0204] Además, el descodificador de vídeo 30 puede usar la imagen descodificada como una imagen de referencia para una imagen que se va a descodificar posteriormente. Es decir, el descodificador de vídeo 30 puede descodificar el valor de restablecimiento de POC de la imagen descodificada como información de movimiento (por ejemplo, un parámetro de movimiento) para un bloque de una imagen posterior (190). La imagen posterior puede ser una imagen posterior en la segunda capa o una imagen de una capa diferente (por ejemplo, una imagen en la misma unidad de acceso que la imagen descodificada). En cualquier caso, la información de movimiento descodificada puede incluir una referencia al valor de POC de la imagen descodificada, que corresponde al valor de restablecimiento de POC para la imagen descodificada (es decir, no el valor de POC para la imagen antes del restablecimiento). A partir de este valor de POC descodificado como información de movimiento para el bloque de la imagen posterior, el descodificador de vídeo 30 puede descodificar el bloque de la imagen posterior en relación con la imagen descodificada (192).

[0205] De esta manera, el procedimiento de la FIG. 6 representa un ejemplo de un procedimiento que incluye determinar si un valor para un elemento sintáctico indica que al menos una porción de un valor de contador del orden de imágenes (POC) de una imagen se debe restablecer en un valor de cero, cuando el valor del elemento sintáctico indica que la porción del valor de POC se debe restablecer en el valor de cero, restablecer al menos la porción del valor de POC de manera que la porción del valor de POC sea igual a cero y descodificar datos de vídeo utilizando el valor de restablecimiento de POC.

[0206] En uno o más ejemplos, las funciones descritas se pueden implementar en hardware, software, firmware o cualquier combinación de los mismos. Si se implementan en software, las funciones se pueden almacenar en, o transmitir por, un medio legible por ordenador, como una o más instrucciones o código, y ejecutar mediante una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que corresponden a un medio tangible tal como unos medios de almacenamiento de datos, o unos medios de comunicación que incluyen cualquier medio que facilita la transferencia de un programa informático de un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por ordenador pueden corresponder, en general, a (1) medios de almacenamiento tangibles legibles por ordenador que son no transitorios o (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible al que se puede acceder desde uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para la implementación de las técnicas descritas en esta divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

[0207] A modo de ejemplo, y no de limitación, dichos medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento en disco óptico, almacenamiento en disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que se puede usar para almacenar código de programa deseado en forma de instrucciones o estructuras de datos y al que se puede

acceder mediante un ordenador. Además, cualquier conexión recibe adecuadamente la denominación de medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde un sitio web, un servidor u otro origen remoto usando un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado digital (DSL) o unas tecnologías inalámbricas tales como infrarrojos, radio y microondas, entonces el cable coaxial, el cable de fibra óptica, el par trenzado, la DSL o las tecnologías inalámbricas tales como infrarrojos, radio y microondas están incluidos en la definición de medio. Sin embargo, se deberá entender que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino que, en cambio, están dirigidos a medios de almacenamiento no transitorio tangibles. El término disco, como se usa en el presente documento, incluye el disco compacto (CD), disco láser, disco óptico, disco versátil digital (DVD), disco flexible y disco Blu-ray, de los cuales los discos flexibles normalmente reproducen datos magnéticamente, mientras que los demás discos reproducen datos ópticamente con láseres. Las combinaciones de los anteriores deberían estar incluidos también dentro del alcance de los medios legibles por ordenador.

[0208] Las instrucciones pueden ser ejecutadas por 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 *in situ* (FPGA) u otra circuitería lógica integrada o discreta equivalente. En consecuencia, el término "procesador", como se usa en el presente documento, se puede referir a cualquier estructura anterior o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento se puede proporcionar dentro de módulos de hardware y/o de software dedicados configurados para codificar y descodificar, o incorporar en un códec combinado. Asimismo, las técnicas se podrían implementar por completo en uno o más circuitos o elementos lógicos.

[0209] Las técnicas de esta divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, incluyendo un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). En esta divulgación se describen diversos componentes, módulos o unidades para destacar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no se requiere necesariamente su realización por diferentes unidades de hardware. En su lugar, como se ha descrito anteriormente, diversas unidades se pueden combinar en una unidad de hardware de códec o proporcionar mediante un grupo de unidades de hardware interoperativas, que incluye uno o más procesadores como se describe anteriormente, junto con software y/o firmware adecuados.

[0210] Se han descrito diversos ejemplos. Estos y otros ejemplos están dentro del alcance de las siguientes reivindicaciones.

REIVINDICACIONES

1. Un procedimiento de descodificación de datos de vídeo, el procedimiento que comprende:

5 recibir una unidad de acceso que incluye un conjunto de primeras imágenes en las primeras capas de codificación de vídeo respectivas y una segunda imagen en una segunda capa de codificación de vídeo diferente de las primeras capas de codificación de vídeo, en el que el conjunto de primeras imágenes incluye todas las imágenes en la unidad de acceso que no son imágenes de actualización instantánea del descodificador, IDR, y la segunda imagen es una imagen IDR; y,

10 para cada primera imagen del conjunto de primeras imágenes:

15 i) recibir un elemento sintáctico indicativo de si al menos una porción del valor de contador del orden de imágenes, POC, de la primera imagen se debe restablecer en un valor de cero,

ii) cuando el valor para el elemento sintáctico indica que la porción del valor de POC se debe restablecer en el valor de cero, restablecer al menos la porción del valor de POC de manera que la porción del valor de POC sea igual a cero, en el que, antes de restablecer el valor de POC, el valor de POC para la primera imagen es diferente de un segundo valor de POC para la segunda imagen, y

20 iii) descodificar datos de vídeo usando el valor de restablecimiento de POC,

en el que un POC es una variable asociada con una imagen que identifica de forma única la imagen asociada entre imágenes en una secuencia de vídeo codificada, y

25 en el que una unidad de acceso incluye todas las imágenes codificadas asociadas con el mismo tiempo de salida.

30 2. El procedimiento de la reivindicación 1, en el que el valor para el elemento sintáctico indica si todos los bits, incluida la porción del valor de POC que debe restablecerse al valor de cero, el procedimiento que además comprende, cuando el valor para el elemento sintáctico indica que todos los bits del valor de POC se deben restablecer al valor cero, restablecer el valor de POC de manera que todos los bits del valor de POC sean iguales a cero.

35 3. El procedimiento de la reivindicación 1, en el que restablecer el valor de POC para la primera imagen comprende restablecer el valor de POC para la primera imagen de manera que, después del restablecimiento, el valor de POC para la primera imagen sea igual al valor de POC para la segunda imagen.

40 4. El procedimiento de la reivindicación 1, en el que la unidad de acceso incluye además una tercera imagen en una tercera capa de codificación de vídeo diferente de la primera capa de codificación de vídeo y la segunda capa de codificación de vídeo, y en el que la descodificación de datos de vídeo que usa el valor de restablecimiento de POC comprende descodificar un bloque de la tercera imagen usando predicción entre capas en relación con la primera imagen cuando el bloque incluye una referencia al valor de restablecimiento de POC.

45 5. El procedimiento de la reivindicación 1, que comprende además, para cada primera imagen del conjunto de primeras imágenes, cuando el elemento sintáctico indica que la porción del valor de POC de la primera imagen se debe restablecer en el valor de cero:

50 determinar un valor de POC de una imagen adicional, en el que la primera imagen y la imagen adicional están en una capa de codificación de vídeo común y una secuencia de vídeo codificada común;

determinar una primera diferencia entre el valor de POC de la primera imagen y el valor de POC de la imagen adicional; y

55 reducir el valor de POC de la imagen adicional de manera que una segunda diferencia entre el valor de POC disminuido de la imagen adicional y el valor de restablecimiento de POC de la primera imagen sea igual a la primera diferencia.

60 6. El procedimiento de la reivindicación 1, que comprende además, para cada primera imagen del conjunto de primeras imágenes, cuando el elemento sintáctico indica que la porción del valor de POC de la primera imagen se debe restablecer en el valor de cero:

determinar los valores de POC para una pluralidad de otras imágenes en una capa de codificación de vídeo común con la primera imagen;

65 determinar las diferencias entre el valor de POC de la primera imagen y los valores de POC para las otras imágenes; y

disminuir los valores de POC para las otras imágenes de manera que las diferencias respectivas entre los valores de POC disminuidos y el valor de restablecimiento de POC de la primera imagen sean iguales a las diferencias determinadas respectivas entre el valor de POC de la primera imagen y los valores de POC para las otras imágenes.

7. El procedimiento de la reivindicación 6, en el que los datos para un conjunto de imágenes de referencia indican si las otras imágenes son imágenes de referencia a corto plazo o imágenes de referencia a largo plazo, y en el que la disminución de los valores de POC comprende:

disminuir los valores de POC para las imágenes de referencia a corto plazo; y

disminuir los valores de POC para las imágenes de referencia a largo plazo.

8. El procedimiento de la reivindicación 1, en el que el elemento sintáctico comprende un indicador de restablecimiento de POC.

9. El procedimiento de la reivindicación 1, que comprende además descodificar una cabecera de fragmento que incluye el elemento sintáctico.

10. El procedimiento de la reivindicación 1, en el que la porción comprende los bits más significativos, MSB, del valor de POC.

11. Un procedimiento de codificación de datos de vídeo, el procedimiento que comprende:

formar una unidad de acceso que incluye un conjunto de primeras imágenes en las primeras capas de codificación de vídeo respectivas y una segunda imagen en una segunda capa de codificación de vídeo, en el que la segunda capa de codificación de vídeo es diferente de las primeras capas de codificación de vídeo, en el que el conjunto de primeras imágenes incluye todas imágenes en la unidad de acceso que no son imágenes de actualización instantánea del descodificador, IDR, y la segunda imagen es una imagen IDR; y,

para cada primera imagen del conjunto de primeras imágenes:

i) determinar si se restablece al menos una porción del valor de contador del orden de imágenes, POC, de la primera imagen en un valor de cero,

ii) en respuesta a la determinación de restablecer al menos la porción del valor de POC, restablecer al menos la porción del valor de POC, de manera que la porción del valor de POC sea igual a cero, en el que, antes de restablecer el valor de POC, el valor de POC para la primera imagen es diferente de un segundo valor de POC para la segunda imagen,

iii) codificar un valor para un elemento sintáctico que indica que al menos la porción del valor de POC se debe restablecer en el valor de cero, y

iv) codificar datos de vídeo utilizando el valor de restablecimiento de POC,

en el que un POC es una variable asociada con una imagen que identifica de forma única la imagen asociada entre imágenes en una secuencia de vídeo codificada, y

en el que una unidad de acceso incluye todas las imágenes codificadas asociadas con el mismo tiempo de salida.

12. Un dispositivo para procesar datos de vídeo, el dispositivo que comprende: uno o más procesadores configurados para:

recibir una unidad de acceso que incluye un conjunto de primeras imágenes en las primeras capas de codificación de vídeo respectivas y una segunda imagen en una segunda capa de codificación de vídeo diferente de la primera capa de codificación de vídeo, en el que el conjunto de primeras imágenes incluye todas las imágenes en la unidad de acceso que no son imágenes de actualización instantánea del descodificador, IDR, y la segunda imagen es una imagen IDR; y,

para cada primera imagen del conjunto de primeras imágenes:

i) recibir un elemento sintáctico indicativo de si al menos una porción del valor de contador del orden de imágenes, POC, de la primera imagen se debe restablecer en un valor de cero;

ii) cuando el valor para el elemento sintáctico indica que la porción del valor de POC se debe restablecer en el valor de cero, restablecer al menos la porción del valor de POC de manera que la porción del valor de POC sea igual a cero, en el que, antes de restablecer el valor de POC, el valor de POC para la primera imagen es diferente de un segundo valor de POC para la segunda imagen, y

5
iii) descodificar datos de vídeo usando el valor de restablecimiento de POC,
en el que un POC es una variable asociada con una imagen que identifica de forma única la imagen asociada entre imágenes en una secuencia de vídeo codificada, y

10
en el que una unidad de acceso incluye todas las imágenes codificadas asociadas con el mismo tiempo de salida.

15 **13.** Un dispositivo para codificar datos de vídeo, el dispositivo que comprende: uno o más procesadores configurados para:

20 formar una unidad de acceso que incluye un conjunto de primeras imágenes en las primeras capas de codificación de vídeo respectivas y una segunda imagen en una segunda capa de codificación de vídeo, en el que la segunda capa de codificación de vídeo es diferente de las primeras capas de codificación de vídeo, en el que el conjunto de primeras imágenes incluye todas imágenes en la unidad de acceso que no son una imágenes de actualización instantánea del descodificador, IDR, y la segunda imagen es una imagen IDR; y,

25 para cada primera imagen del conjunto de primeras imágenes:

25 i) determinar si se restablece al menos una porción del valor de contador del orden de imágenes, POC, de la primera imagen en un valor de cero,

30 ii) en respuesta a la determinación de restablecer al menos la porción del valor de POC, restablecer al menos la porción del valor de POC, de manera que la porción del valor de POC sea igual a cero, en el que, antes de restablecer el valor de POC, el valor de POC para la primera imagen es diferente de un segundo valor de POC para la segunda imagen,

35 iii) codificar un valor para un elemento sintáctico que indica que al menos la porción del valor de POC se debe restablecer en el valor de cero, y

40 iv) codificar datos de vídeo usando el valor de restablecimiento de POC,

en el que un POC es una variable asociada con una imagen que identifica de forma única la imagen asociada entre imágenes en una secuencia de vídeo codificada, y

40 en el que una unidad de acceso incluye todas las imágenes codificadas asociadas con el mismo tiempo de salida.

45 **14.** El dispositivo de la reivindicación 12 o la reivindicación 13, en el que el dispositivo comprende al menos uno de entre:

un circuito integrado;

50 un microprocesador; y

un dispositivo de comunicación inalámbrica.

55 **15.** Un medio de almacenamiento legible por ordenador que tiene instrucciones almacenadas en el mismo que, cuando se ejecutan por un procesador de un dispositivo para codificar o descodificar datos de vídeo, hacen que dicho procesador lleve a cabo un procedimiento de acuerdo con cualquiera de las reivindicaciones 1 a 11.

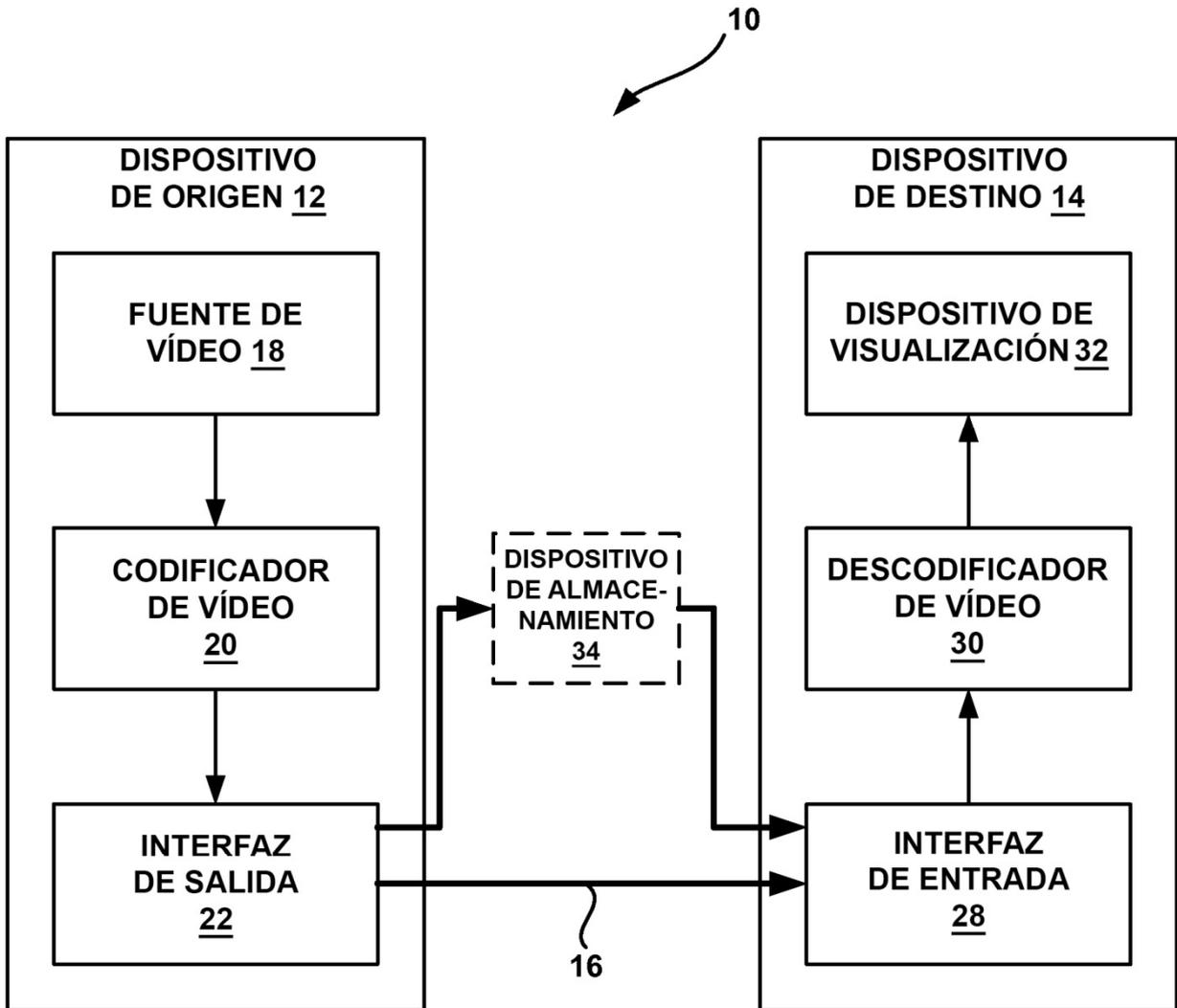


FIG. 1

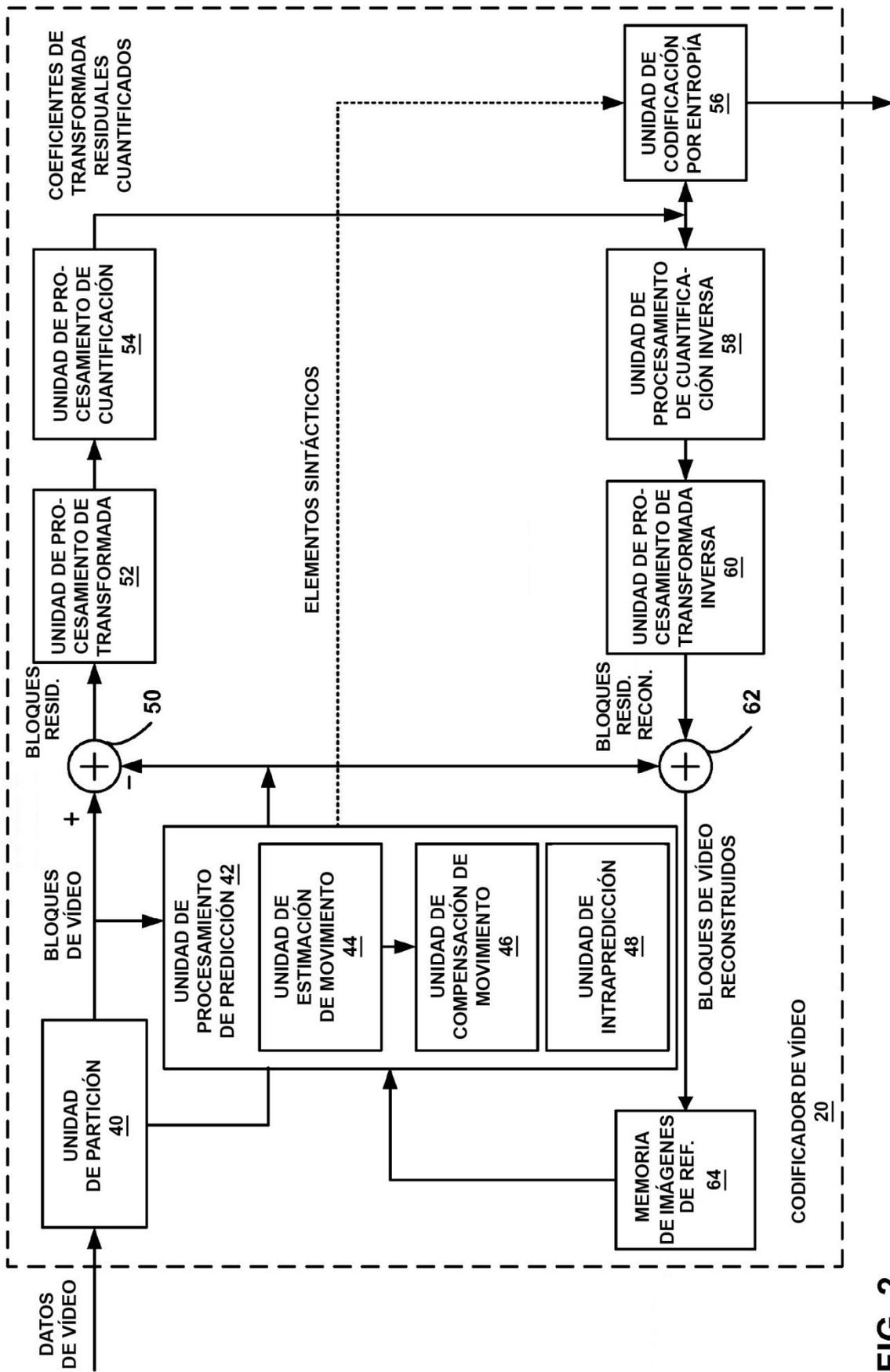


FIG. 2

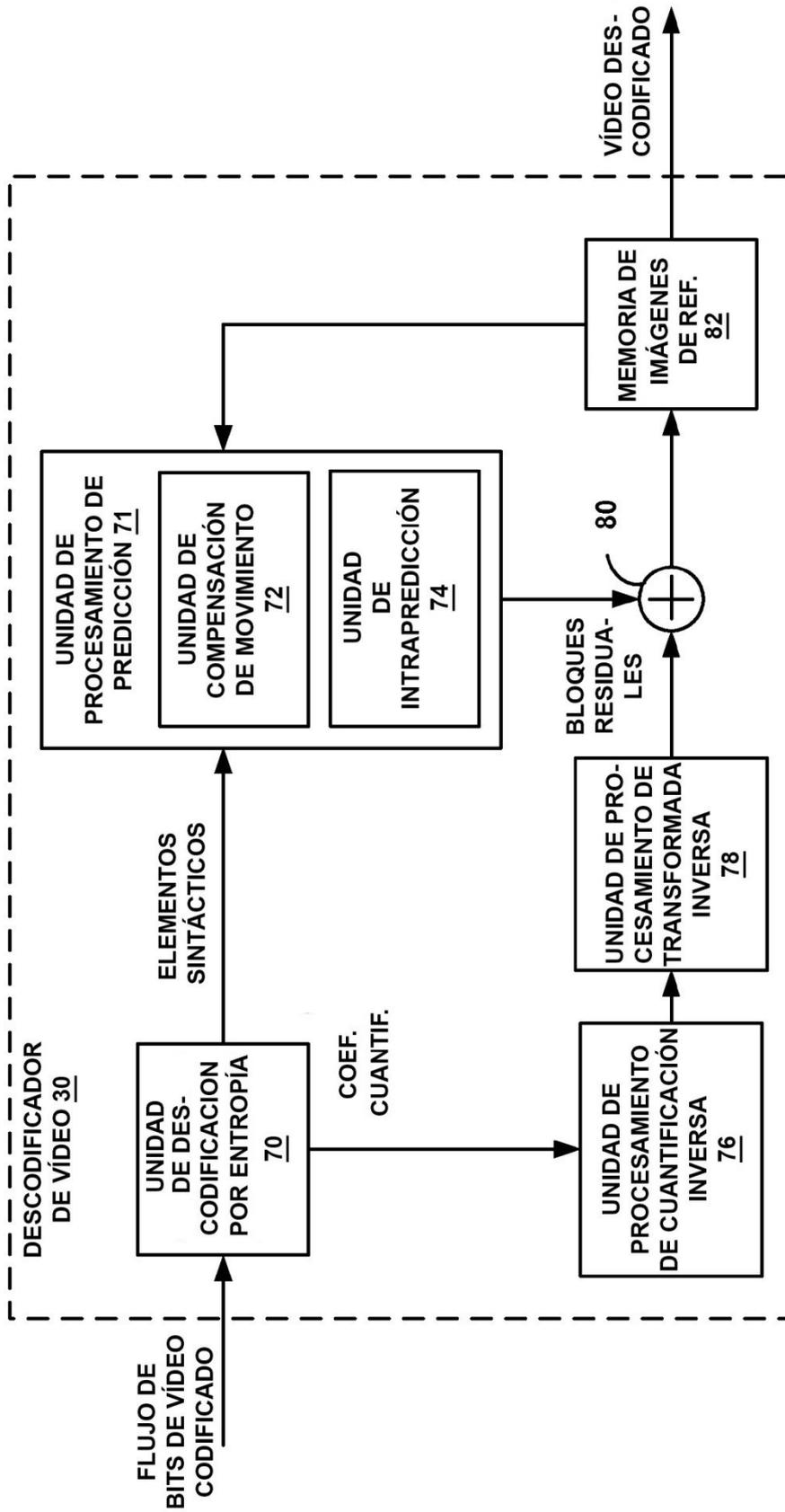


FIG. 3

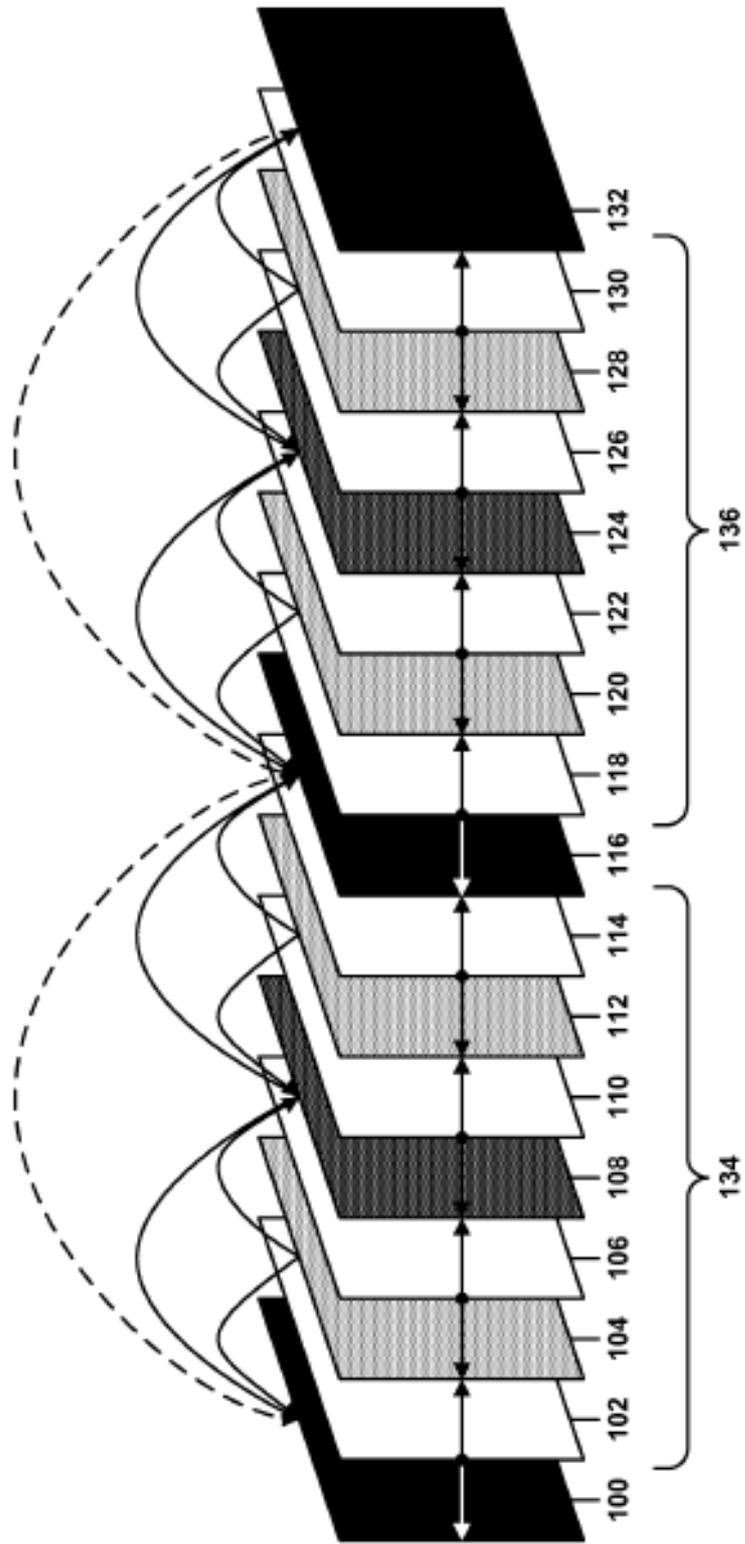


FIG. 4

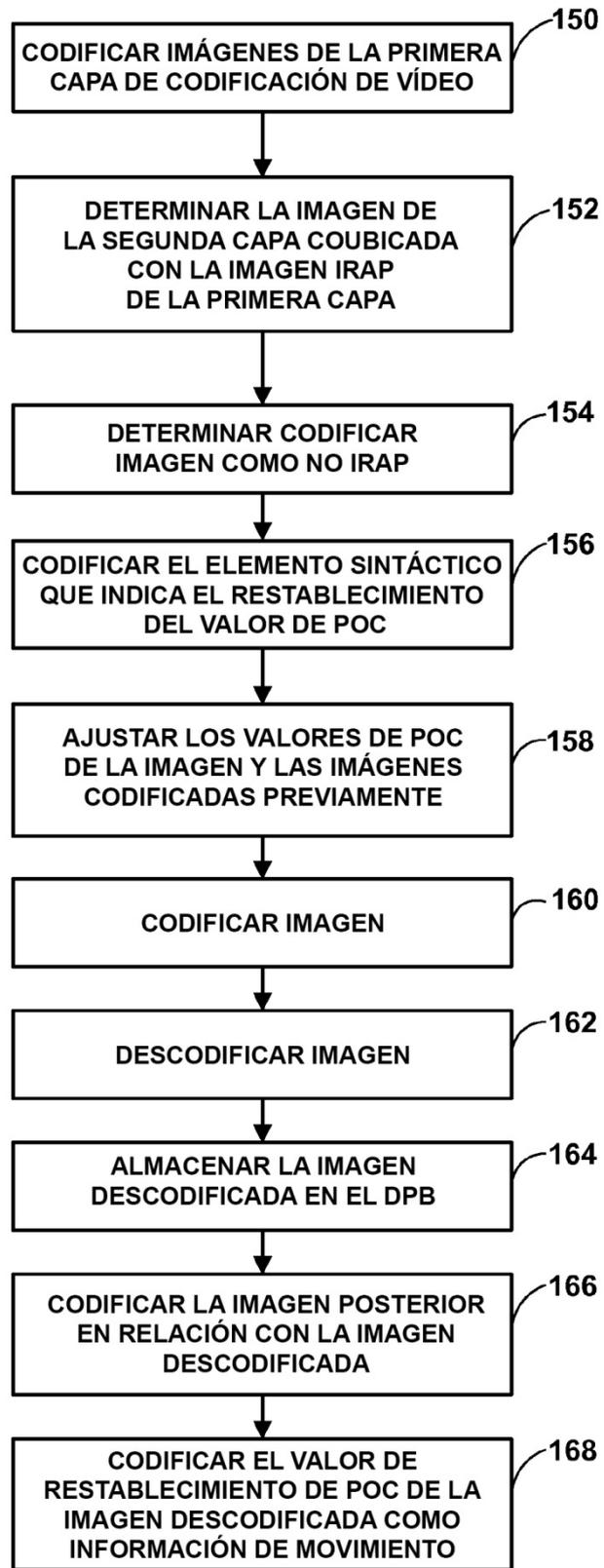


FIG. 5

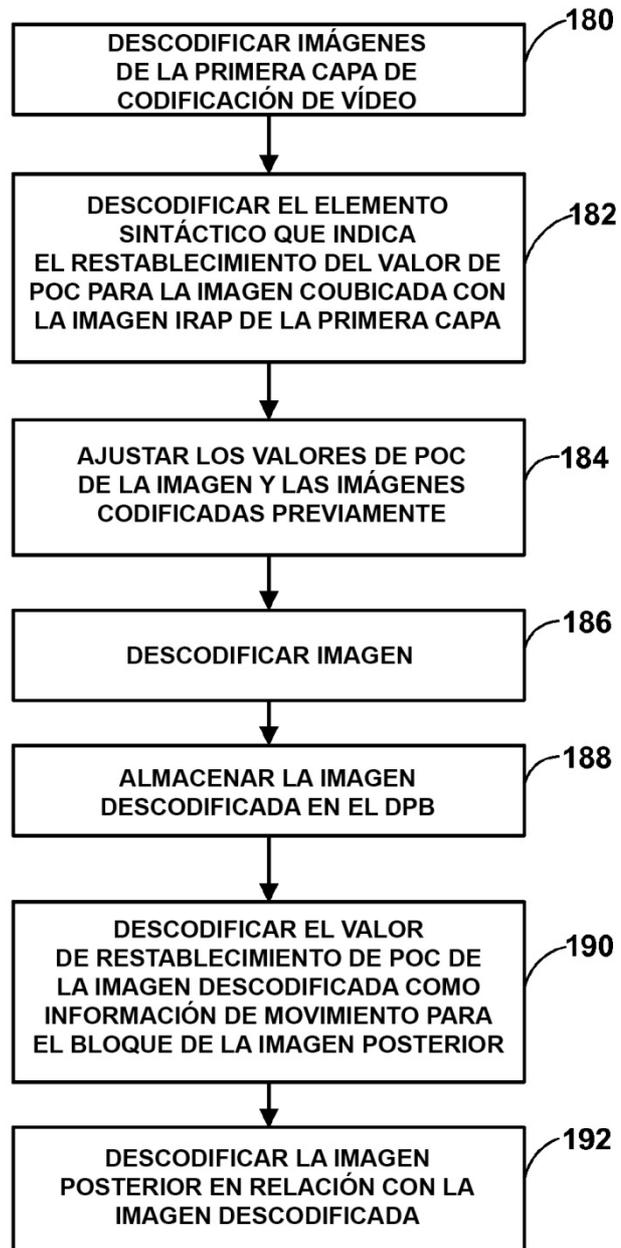


FIG. 6