

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 698 554**

51 Int. Cl.:

H04N 19/463 (2014.01)

H04N 19/58 (2014.01)

H04N 19/423 (2014.01)

H04N 19/70 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **31.10.2012 PCT/US2012/062830**

87 Fecha y número de publicación internacional: **10.05.2013 WO13067033**

96 Fecha de presentación y número de la solicitud europea: **31.10.2012 E 12787601 (9)**

97 Fecha y número de publicación de la concesión europea: **22.08.2018 EP 2774365**

54 Título: **Acceso aleatorio con gestión avanzada de memoria intermedia de imágenes codificadas en codificación de vídeo**

30 Prioridad:

31.10.2011 US 201161553802 P

06.02.2012 US 201261595605 P

30.10.2012 US 201213664279

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
05.02.2019

73 Titular/es:

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

72 Inventor/es:

**CHEN, YING;
WANG, YE-KUI y
CHEN, JIANLE**

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 698 554 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

Acceso aleatorio con gestión avanzada de memoria intermedia de imágenes codificadas en codificación de vídeo

5 **CAMPO TÉCNICO**

[0001] Esta divulgación se refiere a la codificación de vídeo y, más particularmente, a las tramas de codificación de datos de vídeo generadas por procesos de codificación de vídeo.

10 **ANTECEDENTES**

[0002] Las capacidades del vídeo digital pueden incorporarse en una amplia gama de dispositivos, incluyendo televisores digitales, sistemas de radiodifusión directa digital, sistemas de radiodifusión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, ordenadores tipo tablet, lectores de libros electrónicos, cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de videojuegos, consolas 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 digitales implementan técnicas de compresión de vídeo, tales como las descritas en las normas definidas por MPEG-2, MPEG-4, ITU-TH.263, ITU-T H.264/MPEG-4, Parte 10, Codificación Avanzada de Vídeo (AVC), la norma de Codificación de Vídeo de Alta Eficiencia (HEVC) actualmente en desarrollo y las ampliaciones de dichas normas. Los dispositivos de vídeo pueden transmitir, recibir, codificar, descodificar y/o almacenar información de vídeo digital de forma más eficiente, implementando dichas técnicas de compresión de vídeo.

[0003] Las técnicas de compresión de vídeo realizan la predicción espacial (intra-imagen) y/o la predicción temporal (entre imágenes) 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 parte de una trama de vídeo) puede dividirse en bloques de vídeo, que también pueden denominarse bloques de árbol, unidades de codificación (CU) y/o nodos de codificación. Los bloques de vídeo en un fragmento intra-codificado (I) de una imagen se codifican mediante predicción espacial con respecto a muestras de referencia en bloques contiguos de la misma imagen. Los bloques de vídeo en un fragmento inter-codificado (P o B) de una imagen pueden usar la predicción espacial con respecto a muestras de referencia en bloques contiguos de la misma imagen, o la predicción temporal con respecto a muestras de referencia en otras imágenes de referencia. Las imágenes pueden denominarse tramas, y las imágenes de referencia pueden denominarse tramas de referencia.

[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 inter-codificado 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 intra-codificado se codifica de acuerdo con un modo de intra-codificación y los datos residuales. Para una mayor compresión, los datos residuales pueden transformarse desde el dominio de los píxeles al dominio de las transformaciones, dando como resultado unos coeficientes de transformación residuales, que posteriormente se pueden cuantificar. Los coeficientes de transformación cuantificados, inicialmente dispuestos en una formación bidimensional, pueden escanearse en orden con el fin de producir un vector unidimensional de coeficientes de transformación. A continuación se puede aplicar codificación de entropía para lograr aún más compresión.

[0005] En el documento US 2011/0013889 A1 de Wu y col., se divulga un flujo de bits de vídeo con imágenes que comprenden el contenido inter-codificado que puede descodificarse al recibir una instrucción de inicio de canal o de búsqueda de archivo. Las imágenes para la descodificación inicial y la visualización del flujo de bits se pueden seleccionar basándose al menos en parte en uno o más parámetros de ajuste que establecen una preferencia entre una latencia de comienzo para mostrar el vídeo y posibles defectos en el vídeo mostrado. En algunos modos de realización, para implementar la descodificación sobre un inicio de canal o búsqueda de archivo, se generan uno o más tipos de datos para una o más imágenes. Por ejemplo, los recuentos de orden de imagen se generan para las imágenes después de un inicio de canal o una operación de búsqueda de archivo. Como otro ejemplo, un descodificador genera un valor de número de trama que activa la reinicialización de una memoria intermedia de imágenes de referencia antes de la descodificación después de una operación de inicio de canal o búsqueda de archivo.

[0006] En "Study of Final Committee Draft of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC)" ["Estudio del borrador del comité final de la especificación conjunta de vídeo (Rec. ITU-T H.264 | ISO/IEC 14496-10 AVC)"] de Equipo de vídeo conjunto (JVT) de ISO/IEC MPEG e ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 e ITU-T SG16 Q.6) Documento JVT-F100 6.^a reunión : Awaji Island, JP, 5-13 de diciembre de 2002, se presenta el borrador del estudio de salida de la reunión de Awaji de la JVT, producida el 16 de febrero de 2003. Tiene el estado de "Estudio del Borrador del Comité Final" en WG11 y del borrador de un grupo de ponentes en ITU-T SG16 Q.6. Como parte de la divulgación, un "punto de recuperación" se define como un punto en la secuencia de

unidad NAL en el que se logra la recuperación de una representación exacta o aproximada de las imágenes descodificadas representadas por la secuencia de unidad NAL después de un acceso aleatorio o un enlace roto. También se divulga una tabla de sintaxis de mensajes SEI de período de almacenamiento en memoria intermedia, que enumera los elementos sintácticos `initial_cpb_removal_delay`.

5

SUMARIO

[0007] Esta divulgación describe técnicas para el acceso aleatorio en la codificación de vídeo. En particular, la divulgación describe varias técnicas para codificar secuencias de vídeo que incluyen una o más tramas, o "imágenes", en el que una primera imagen codificada de una secuencia de vídeo codificada (CVS) particular en un flujo de bits conforme puede ser una imagen de punto de acceso aleatorio (RAP) que no es una imagen de actualización de descodificación instantánea (IDR). Por ejemplo, de acuerdo con las técnicas, la primera imagen codificada puede ser una imagen de acceso aleatorio limpio (CRA).

10

[0008] La invención está definida en las reivindicaciones adjuntas, a las cuales se hará referencia a continuación.

15

[0009] Como un ejemplo, las técnicas de esta divulgación pueden permitir que un descodificador de vídeo se ajuste a las técnicas para descodificar con éxito un flujo de bits comenzando a partir de una imagen RAP no IDR de una manera predecible y definida, o "estándar". Por ejemplo, las técnicas divulgadas pueden permitir que el descodificador de vídeo conforme manipule diversas propiedades de salida y referencia de las denominadas "imágenes principales" asociadas con la primera imagen codificada que también están incluidas en el flujo de bits. Como resultado, las técnicas pueden permitir el acceso aleatorio relativamente mejorado del flujo de bits mediante el descodificador de vídeo, en comparación con otras técnicas. Por ejemplo, las técnicas pueden facilitar un acceso aleatorio "más fino" o más granular del flujo de bits habilitando el descodificador de vídeo para descodificar el flujo de bits en relativamente más puntos de inicio, o imágenes de acceso (es decir, imágenes no IDR) del flujo de bits, en comparación con otras técnicas (por ejemplo, técnicas que permiten el acceso aleatorio de un flujo de bits solo desde imágenes IDR). Adicionalmente, las técnicas pueden permitir que el descodificador de vídeo conformador mejore la calidad visual de una o más imágenes también incluidas en el flujo de bits, por ejemplo, evitando emitir y/o usar como imágenes de referencia las imágenes principales asociadas con la primera imagen.

20

25

30

[0010] De forma alternativa, como otro ejemplo, las técnicas divulgadas pueden permitir que un codificador de vídeo que se ajuste a las técnicas genere un flujo de bits que excluya las imágenes principales asociadas con una primera imagen codificada del flujo de bits que es una imagen RAP no IDR. Como resultado, un descodificador de vídeo que también se ajusta a las técnicas divulgadas puede descodificar con éxito el flujo de bits de una manera predecible y definida.

35

[0011] En consecuencia, el uso de las técnicas de esta divulgación puede mejorar la interoperabilidad de los sistemas y dispositivos de codificación y descodificación de vídeo, y la experiencia del usuario, en general, para el acceso aleatorio de flujo de bits que puede ocurrir frecuentemente en diversas aplicaciones de vídeo.

40

[0012] En un ejemplo de la divulgación, un procedimiento de descodificación de datos de vídeo incluye recibir un flujo de bits que comprende una o más imágenes de una CVS, descodificar una primera imagen de la una o más imágenes de acuerdo con un orden de descodificación asociado con la CVS, en el que la primera imagen es una imagen RAP que no es una imagen IDR, y descodificar al menos una de las una o más imágenes, distintas de la primera imagen, siguiendo a la primera imagen de acuerdo con el orden de descodificación, basándose en la primera imagen descodificada.

45

[0013] En otro ejemplo de la divulgación, un procedimiento de codificación de datos de vídeo incluye generar un flujo de bits que comprende una o más imágenes de una CVS, en el que una primera imagen de la una o más imágenes de acuerdo con un orden de descodificación asociado con la CVS es una imagen RAP que no es una imagen IDR, en el que generar el flujo de bits comprende evitar incluir al menos una de las una o más imágenes, distintas de la primera imagen, que corresponde a una imagen principal asociada con la primera imagen, en el flujo de bits, en el que la imagen principal comprende una imagen que sigue a la primera imagen de acuerdo con el orden de descodificación y precede a la primera imagen de acuerdo con un orden de visualización asociado con la CVS, y en el que la primera imagen es descodificable, y en el que al menos una de las una o más imágenes, además de la primera imagen, después de la primera imagen de acuerdo con el orden de descodificación, se puede descodificar basándose en la primera imagen.

50

55

[0014] En otro ejemplo de la divulgación, un aparato configurado para descodificar datos de vídeo incluye un descodificador de vídeo configurado para recibir un flujo de bits que comprende una o más imágenes de una CVS, descodificar una primera imagen de la una o más imágenes de acuerdo con un orden de descodificación asociado con la CVS, en el que la primera imagen es una imagen RAP que no es una imagen IDR, y descodificar al menos una de las una o más imágenes, distintas de la primera imagen, que siguen a la primera imagen de acuerdo con el orden de descodificación, basándose en la primera imagen descodificada.

60

65

5 [0015] En otro ejemplo de la divulgación, un aparato configurado para codificar datos de vídeo incluye un codificador de vídeo configurado para generar un flujo de bits que comprende una o más imágenes de una CVS, en el que una primera imagen de la una o más imágenes de acuerdo con un orden de descodificación asociado con la CVS es una imagen RAP que no es una imagen IDR, en el que para generar el flujo de bits, el codificador de vídeo está configurado para evitar incluir al menos una de las una o más imágenes, que no sea la primera, que corresponde a una imagen principal asociada con la primera imagen, en el flujo de bits, en el que la imagen principal comprende una imagen que sigue a la primera imagen de acuerdo con el orden de descodificación y precede a la primera imagen de acuerdo con un orden de visualización asociado con la CVS, y en el que la primera imagen es descodificable, y en el que al menos una de las una o más imágenes, aparte de la primera imagen, que sigue a la primera imagen de acuerdo con el orden de descodificación, es descodificable basándose en la primera imagen.

15 [0016] En otro ejemplo de la divulgación, un dispositivo para descodificar datos de vídeo incluye medios para recibir un flujo de bits que comprende una o más imágenes de una CVS, medios para descodificar una primera imagen de la una o más imágenes de acuerdo con un orden de descodificación asociado con la CVS, donde la primera imagen es una imagen RAP que no es una imagen IDR, y medios para descodificar al menos una de las una o más imágenes, distintas de la primera imagen, que siguen a la primera imagen de acuerdo con el orden de descodificación, basándose en la primera descodificación imagen.

20 [0017] En otro ejemplo de la divulgación, un dispositivo para codificar datos de vídeo incluye medios para generar un flujo de bits que comprende una o más imágenes de una CVS, en el que una primera imagen de la una o más imágenes de acuerdo con un orden de descodificación asociado con la CVS es una imagen RAP que no es una imagen IDR, en el que los medios para generar el flujo de bits comprenden medios para evitar incluir al menos una de las una o más imágenes, distintas de la primera imagen, que corresponde a una imagen principal asociada con la primera imagen, en el flujo de bits, en el que la imagen principal comprende una imagen que sigue a la primera imagen de acuerdo con el orden de descodificación y precede a la primera imagen de acuerdo con un orden de visualización asociado con la CVS, y en el que la primera imagen es descodificable, y en el que al menos una o más imágenes, distintas de la primera imagen, que siguen a la primera imagen de acuerdo con el orden de descodificación, se pueden descodificar basándose en la primera imagen.

30 [0018] Las técnicas descritas en esta divulgación pueden implementarse en hardware, software, firmware o cualquier combinación de estos. Si se implementa en hardware, un aparato puede realizarse como un circuito integrado, un procesador, lógica discreta o cualquier combinación de los mismos. Si se implementa en software, el software puede ejecutarse en uno o más procesadores, tales como un microprocesador, un circuito integrado específico de la aplicación (ASIC), una matriz de puertas programables in situ (FPGA) o un procesador de señales digitales (DSP). El software que ejecuta las técnicas se puede almacenar inicialmente en un medio legible por ordenador tangible y cargarse y ejecutarse en el procesador.

40 [0019] Por consiguiente, en otro ejemplo, esta divulgación contempla un medio de almacenamiento legible por ordenador que almacena instrucciones que, cuando se ejecutan, hacen que uno o más procesadores reciban un flujo de bits que comprende una o más imágenes de una CVS, descodifica una primera imagen de la una o más imágenes de acuerdo con un orden de descodificación asociado con la CVS, en el que la primera imagen es una imagen RAP que no es una imagen IDR, y descodifica al menos una de las una o más imágenes, distintas de la primera imagen, siguiendo a la primera imagen de acuerdo con el orden de descodificación, basándose en la primera imagen descodificada.

50 [0020] En otro ejemplo, esta divulgación contempla un medio de almacenamiento legible por ordenador que almacena instrucciones que, cuando se ejecutan, hacen que uno o más procesadores generen un flujo de bits que comprende una o más imágenes de una CVS, en el que una primera imagen de una o más imágenes de acuerdo con un orden de descodificación asociado con la CVS es una imagen RAP que no es una imagen IDR, en el que las instrucciones que hacen que uno o más procesadores generen el flujo de bits comprenden instrucciones que hacen que uno o más procesadores eviten incluir al menos una de las una o más imágenes, distintas de la primera imagen, que corresponden a una imagen principal asociada con la primera imagen, en el flujo de bits, en el que la imagen principal comprende una imagen que sigue a la primera imagen de acuerdo con el orden de descodificación y precede a la primera imagen de acuerdo con un orden de visualización asociada con la CVS, y en la que la primera imagen es descodificable, y en la que al menos una de las una o más imágenes, distinta de la primera imagen, después de la primera imagen de acuerdo con el orden de descodificación, se puede descodificar basándose en la primera imagen.

60 [0021] Los detalles de uno o más ejemplos se exponen en los dibujos adjuntos y en la siguiente descripción. Otras características, objetivos y ventajas resultarán evidentes a partir de la descripción y de los dibujos, y a partir de las reivindicaciones.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

65 [0022]

La FIG. 1 es un diagrama de bloques que ilustra un ejemplo de un sistema de codificación y decodificación de vídeo que puede realizar técnicas para el acceso aleatorio con gestión avanzada de memoria intermedia de imágenes de decodificador (DPB), coherente con las técnicas de esta divulgación.

5 La FIG. 2 es un diagrama de bloques que ilustra un ejemplo de un codificador de vídeo que puede implementar las técnicas para acceso aleatorio con gestión de DPB avanzada, coherentes con las técnicas de esta divulgación.

10 La FIG. 3 es un diagrama de bloques que ilustra un ejemplo de un decodificador de vídeo que puede implementar las técnicas para acceso aleatorio con gestión de DPB avanzada, de forma coherente con las técnicas de esta divulgación.

15 La FIG. 4 es un diagrama conceptual que ilustra un ejemplo de jerarquías de referencia entre imágenes de grupos de imágenes (GOP) de datos de vídeo, de forma coherente con las técnicas de esta divulgación.

La FIG. 5 es un diagrama de flujo que ilustra un procedimiento de ejemplo para realizar un acceso aleatorio de un flujo de bits que incluye una o más imágenes de datos de vídeo mediante un decodificador de vídeo, de acuerdo con las técnicas de esta divulgación.

20 La FIG. 6 es un diagrama de flujo que ilustra un procedimiento de ejemplo para generar un flujo de bits que incluye una o más imágenes de datos de vídeo mediante un codificador de vídeo, de acuerdo con las técnicas de esta divulgación.

DESCRIPCIÓN DETALLADA

25 **[0023]** Esta divulgación describe técnicas para el acceso aleatorio en la codificación de vídeo. En particular, la divulgación describe varias técnicas para codificar secuencias de vídeo que incluyen una o más tramas, o "imágenes", en el que una primera imagen codificada de una secuencia de vídeo codificada (CVS) particular en un flujo de bits conforme puede ser una imagen de punto de acceso aleatorio (RAP) que no es una imagen de actualización de decodificación instantánea (IDR). Por ejemplo, de acuerdo con las técnicas, la primera imagen codificada puede ser una imagen de acceso aleatorio limpio (CRA).

35 **[0024]** Como un ejemplo, las técnicas de esta divulgación pueden permitir que un decodificador de vídeo se ajuste a las técnicas para decodificar con éxito un flujo de bits comenzando a partir de una imagen RAP no IDR de una manera predecible y definida, o "estándar". Por ejemplo, las técnicas divulgadas pueden permitir que el decodificador de vídeo conforme manipule diversas propiedades de salida y referencia de las denominadas "imágenes principales" asociadas con la primera imagen codificada que también están incluidas en el flujo de bits. Como resultado, las técnicas pueden permitir el acceso aleatorio relativamente mejorado del flujo de bits mediante el decodificador de vídeo, en comparación con otras técnicas. Por ejemplo, las técnicas pueden facilitar un acceso aleatorio "más fino" o más granular del flujo de bits habilitando el decodificador de vídeo para decodificar el flujo de bits en relativamente más puntos de inicio, o imágenes de acceso (es decir, imágenes no IDR) del flujo de bits, en comparación con otras técnicas (por ejemplo, técnicas que permiten el acceso aleatorio de un flujo de bits solo desde imágenes IDR). Adicionalmente, las técnicas pueden permitir que el decodificador de vídeo conformador mejore la calidad visual de una o más imágenes también incluidas en el flujo de bits, por ejemplo, evitando emitir y/o usar como imágenes de referencia las imágenes principales asociadas con la primera imagen.

45 **[0025]** De forma alternativa, como otro ejemplo, las técnicas divulgadas pueden permitir que un codificador de vídeo que se ajuste a las técnicas genere un flujo de bits que excluya las imágenes principales asociadas con una primera imagen codificada del flujo de bits que es una imagen RAP no IDR. Como resultado, un decodificador de vídeo que también se ajusta a las técnicas divulgadas puede decodificar con éxito el flujo de bits de una manera predecible y definida.

50 **[0026]** En consecuencia, el uso de las técnicas de esta divulgación puede mejorar la interoperabilidad de los sistemas y dispositivos de codificación y decodificación de vídeo, y la experiencia del usuario, en general, para el acceso aleatorio de flujo de bits que puede ocurrir frecuentemente en diversas aplicaciones de vídeo.

55 **[0027]** Específicamente, las técnicas descritas en el presente documento pueden incluir al menos uno o más de los siguientes aspectos novedosos, en comparación con otras técnicas: (1) detectar cuándo se produce un acceso aleatorio desde una imagen RAP no IDR (por ejemplo, una imagen de CRA); 2) identificar y decodificar una o más imágenes que siguen a la imagen RAP no IDR en un orden de decodificación, pero preceden a la imagen RAP no IDR en un orden de salida (es decir, una o más "imágenes principales" de la imagen RAP no IDR); y (3) especificar que cada una de las una o más imágenes principales de la imagen RAP no IDR no se emite, incluso en el caso de que un output_flag de un elemento sintáctico señalado correspondiente sea igual a verdadero, o "1" (es decir, el output_flag indica que la imagen respectiva se va a emitir), y que la imagen respectiva no se usa como una imagen de referencia para ninguna otra imagen que siga a la imagen RAP no IDR en el orden de decodificación y el orden de salida.

[0028] De esta manera, un flujo de bits que incluye una o más imágenes de datos de vídeo y comienza con una imagen RAP no IDR puede descodificarse de manera predecible y definida mediante un descodificador de vídeo conforme a las técnicas de esta divulgación. De forma alternativa, un codificador de vídeo conforme a las técnicas divulgadas puede generar un flujo de bits que incluye una o más imágenes de datos de vídeo y comienza con una imagen RAP no IDR, de manera que el flujo de bits puede descodificarse de manera predecible y definida mediante un descodificador de vídeo también conforme a las técnicas. Como resultado, puede haber una mejora relativa en la experiencia del usuario cuando se realiza un acceso aleatorio de un flujo de bits que incluye una o más imágenes de datos de vídeo, cuando se usan las técnicas de esta divulgación. En particular, puede haber una mejora relativa en la granularidad del acceso aleatorio, así como en la calidad visual de una o más imágenes del flujo de bits, y/o de una CVS que incluye una o más imágenes como un todo, cuando se usan las técnicas divulgadas.

[0029] La FIG. 1 es un diagrama de bloques que ilustra un ejemplo de un sistema de codificación y descodificación de vídeo que puede realizar técnicas para el acceso aleatorio con gestión avanzada de memoria intermedia de imágenes de descodificador (DPB), coherente con las técnicas de esta divulgación. 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 sobremesa, ordenadores plegables (es decir, portátiles), ordenadores tipo tablet, descodificadores, equipos telefónicos tales como los denominados teléfonos “inteligentes”, los denominados paneles “inteligentes”, televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivos de transmisión continua de vídeo o similares. En algunos casos, el dispositivo de origen 12 y el dispositivo de destino 14 pueden estar equipados para la comunicación inalámbrica.

[0030] El dispositivo de destino 14 puede recibir los datos de vídeo codificados que se van a descodificar mediante un enlace 16. El enlace 16 puede comprender cualquier tipo de medio o dispositivo capaz de desplazar los datos de vídeo codificados desde el dispositivo de origen 12 al dispositivo de destino 14. En un ejemplo, el enlace 16 puede comprender un medio de comunicación para permitir al dispositivo de origen 12 transmitir los datos de vídeo codificados directamente al dispositivo de destino 14 en tiempo real. Los datos de vídeo codificados pueden modularse de acuerdo con una norma de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitirse al dispositivo de destino 14. El medio de comunicación puede comprender cualquier medio de comunicación inalámbrica o alámbrica, tal como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión física. 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 routers, conmutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo de origen 12 al dispositivo de destino 14.

[0031] De forma alternativa, los datos codificados pueden ser emitidos desde la interfaz de salida 22 a un dispositivo de almacenamiento 24. De forma similar, se puede acceder a los datos codificados desde el dispositivo de almacenamiento 24 mediante una interfaz de entrada 26. El dispositivo de almacenamiento 24 puede incluir cualquiera de una diversidad de medios de almacenamiento de datos, de acceso distribuido o local, tales como una unidad de disco duro, unos discos Blu-ray, discos DVD, discos CD-ROM, una memoria flash, memoria volátil o no volátil o cualquier otro medio de almacenamiento digital adecuado, para almacenar datos de vídeo codificados. En un ejemplo adicional, el dispositivo de almacenamiento 24 puede corresponder a un servidor de archivos o a otro dispositivo de almacenamiento en memoria intermedia que pueda retener el vídeo codificado generado por el dispositivo de origen 12. El dispositivo de destino 14 puede acceder a los datos de vídeo almacenados desde el dispositivo de almacenamiento 24 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. Ejemplos de servidores de archivos incluyen un servidor web (por ejemplo, para un sitio web), un servidor FTP, dispositivos de almacenamiento conectado en red (NAS) o una unidad de disco local. El dispositivo de destino 14 puede acceder a los datos de vídeo codificados mediante cualquier conexión de datos estándar, incluyendo una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión Wi-Fi), una conexión alámbrica (por ejemplo, DSL, módem de cable, etc.), o una combinación de ambos que sea adecuada para acceder a datos de vídeo codificados almacenados en un servidor de archivos. La transmisión de datos de vídeo codificados desde el dispositivo de almacenamiento 24 puede ser una transmisión en continuo, una transmisión de descarga o una combinación de ambas.

[0032] Las técnicas de esta divulgación no están limitadas necesariamente a aplicaciones o configuraciones inalámbricas. Las técnicas pueden aplicarse a la codificación de vídeo, en soporte de cualquiera de una diversidad de aplicaciones de multimedia, tales como radiodifusiones de televisión por el aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de vídeo en continuo, por ejemplo, mediante 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 configurarse para admitir la transmisión de vídeo unidireccional o bidireccional para admitir aplicaciones tales como la transmisión continua de vídeo, la reproducción de vídeo, la difusión de vídeo y/o la videotelefonía.

[0033] 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 captura de vídeo, por ejemplo, una videocámara, un archivo de vídeo que contiene vídeo previamente capturado, una interfaz de alimentación de vídeo para recibir vídeo desde un proveedor de contenido de vídeo y/o un sistema de gráficos de ordenador para generar datos de gráficos de ordenador como el vídeo de origen, o una combinación de tales fuentes. Como 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 pueden aplicarse a aplicaciones inalámbricas y/o cableadas.

[0034] El vídeo capturado, precapturado o generado por ordenador puede ser codificado por el codificador de vídeo 20. Los datos de vídeo codificados pueden ser transmitidos directamente al dispositivo de destino 14 mediante la interfaz de salida 22 del dispositivo de origen 12. Los datos de vídeo codificados también (o de forma alternativa) pueden almacenarse en el dispositivo de almacenamiento 24 para un acceso posterior por el dispositivo de destino 14 u otros dispositivos, para su descodificación y/o reproducción.

[0035] El dispositivo de destino 14 incluye una interfaz de entrada 26, un descodificador de vídeo 30 y un dispositivo de visualización 28. En algunos casos, la interfaz de entrada 26 puede incluir un receptor y/o un módem. La interfaz de entrada 26 del dispositivo de destino 14 recibe los datos de vídeo codificado por el enlace 16 o desde el dispositivo de almacenamiento 24. Los datos de vídeo codificados, comunicados por el enlace 16, o proporcionados en el dispositivo de almacenamiento 24, 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 pueden incluirse 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.

[0036] El dispositivo de visualización 28 puede estar integrado con, o ser externo a, el dispositivo de destino 14. En algunos ejemplos, el dispositivo de destino 14 puede incluir un dispositivo de visualización integrado y también estar configurado para interconectarse con un dispositivo de visualización externo. En otros ejemplos, el dispositivo de destino 14 puede ser un dispositivo de visualización. En general, el dispositivo de visualización 28 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.

[0037] El codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con una norma de compresión de vídeo, tal como la norma de Codificación de Vídeo de Alta Eficiencia (HEVC), actualmente en fase de desarrollo por parte del Equipo de Colaboración Conjunta sobre Codificación de vídeo (JCT-VC) del Grupo de Expertos de Codificación de Vídeo (VCEG) de la ITU-T y el Grupo de Expertos en Imágenes en Movimiento (MPEG) de ISO/IEC, y puede estar conforme con el modelo de prueba de HEVC (HM). De forma alternativa, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con otras normas privadas o de la industria, tales como la norma ITU-T H.264, denominada de forma alternativa MPEG-4, Parte 10, AVC, o ampliaciones de dichas normas. Sin embargo, las técnicas de esta divulgación no están limitadas a ninguna norma de codificación particular. Otros ejemplos de normas de compresión de vídeo incluyen MPEG-2 e ITU-T H.263. Un borrador reciente de la norma HEVC, denominado "Borrador 8 de trabajo de la HEVC" o "WD8", se describe en el documento JCTVC-J1003_d7, Bross et al., titulado "High efficiency video coding (HEVC) text specification draft 8" ["Borrador 8 de Memoria descriptiva textual de la Codificación de Vídeo de Alta Eficiencia (HEVC)"], Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, 10.ª reunión: Estocolmo, Suecia, 11-20 de julio de 2012, que a partir del 17 de octubre de 2012 puede descargarse en http://phenix.int-evry.fr/jct/doc_end_user/documents/10_Stockholm/wg11/JCTVC-J1003-v8.zip.

[0038] Otro borrador de la norma HEVC, al que se hace referencia en esta divulgación como "Borrador de trabajo HEVC 4" o "WD4", se describe en el documento JCTVC-F803, Bross et al., "WD4: Working Draft 4 of High-Efficiency Video Coding" ["WD4: Borrador de Trabajo 4 de Codificación de Vídeo de Alta Eficiencia"], Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, 6.ª reunión: Torino, IT, del 14 al 22 de julio de 2011, que, a partir del 17 de octubre de 2012, se puede descargar en http://phenix.int-evry.fr/jct/doc_end_user/documents/6_Torino/wg11/JCTVC-F803-v8.zip.

[0039] Todavía otro borrador de la norma HEVC, denominado en esta divulgación "Borrador de trabajo HEVC 5" o "WD5", se describe en el documento JCTVC-G1103, Bross et al., "WD5: Working Draft 5 of High-Efficiency Video Coding" ["WD5: Borrador de Trabajo 5 de Codificación de Vídeo de Alta Eficiencia"], Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, 7.ª reunión: Ginebra, Suiza, 21-30 de noviembre, 2011, que a partir del 17 de octubre de 2012 puede descargarse en http://phenix.int-evry.fr/jct/doc_end_user/documents/7_Geneva/wg11/JCTVC-G1103-v12.zip.

[0040] Aunque no se muestra en la FIG. 1, en algunos aspectos, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden estar integrados, cada uno de ellos, en 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 diferentes. Si procede, en algunos ejemplos, las unidades MUX-DEMUX pueden conformarse al protocolo de multiplexado ITU H.223 o a otros protocolos, tales como el protocolo de datagramas de usuario (UDP).

[0041] El codificador de vídeo 20 y el descodificador de vídeo 30 pueden implementarse como cualquiera de una variedad de circuitos adecuados de codificadores o descodificadores, 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 mediante uno o más procesadores para realizar las técnicas de esta divulgación. Cada uno del codificador de vídeo 20 y el descodificador de vídeo 30 pueden estar incluidos en uno o más codificadores o descodificadores, cualquiera de los cuales puede estar integrado como parte de un codificador/descodificador ("CÓDEC") combinado en un dispositivo respectivo.

[0042] Los esfuerzos de normalización de HEVC se basan en un modelo en evolución de un dispositivo de codificación de vídeo denominado modelo de prueba de HEVC (HM). El HM supone varias capacidades adicionales de los dispositivos de codificación de vídeo respecto a dispositivos existentes de acuerdo con, por ejemplo, la norma ITU-T H.264/AVC. Por ejemplo, mientras que la norma H.264 proporciona nueve modos de codificación de predicción intra, el HM puede proporcionar hasta treinta y cinco modos de codificación de predicción intra.

[0043] En general, el modelo de funcionamiento del HM describe que una trama o imagen de vídeo puede dividirse en una secuencia de bloques arbolados o unidades de codificación de máximo tamaño (LCU), que incluyen muestras tanto de luma como de croma. Un bloque arbolado tiene un fin similar al de un macrobloque de la norma H.264. Un fragmento incluye un cierto número de bloques arbolados consecutivos en orden de codificación. Una trama o imagen de vídeo puede dividirse en uno o más fragmentos. Cada bloque arbolado puede dividirse en unidades de codificación (CU) de acuerdo con un árbol cuádruple. Por ejemplo, un bloque arbolado, 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 nodo hijo final, no dividido, como 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 arbolado, y también pueden definir un tamaño mínimo de los nodos de codificación.

[0044] Una CU incluye un nodo de codificación y unidades de predicción (PU) y unidades de transformación (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 arbolado, 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 basándose en si la CU está codificada en modo de salto o directo, codificada en modo de predicción intra o codificada en modo de predicción inter. Las PU pueden dividirse para tener 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 cuádruple. Una TU puede tener forma cuadrada o no cuadrada.

[0045] La norma HEVC soporta transformaciones de acuerdo con las TU, que pueden ser diferentes para diferentes CU. El tamaño de las TU típicamente se basa en el tamaño de las PU de una CU dada, definida para una LCU dividida, aunque puede que no siempre sea 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 mediante una estructura de árbol cuádruple conocida como "árbol cuádruple residual" (RQT). Los nodos hoja del RQT pueden denominarse TU. Los valores de diferencias de píxeles, asociados a las TU, pueden transformarse para generar coeficientes de transformación, que pueden cuantificarse.

[0046] En general, una PU incluye datos relacionados con el proceso de predicción. Por ejemplo, cuando la PU está codificada de manera intramodal, la PU puede incluir datos que describan un modo de predicción intra para la PU. Como otro ejemplo, cuando la PU está codificada de modo de inter, 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 un cuarto de píxel o precisión de un octavo de píxel), una imagen de referencia a la que apunta el vector de movimiento y/o una lista de imágenes de referencia (por ejemplo, la Lista 0, la Lista 1 o la Lista C) para el vector de movimiento.

[0047] En general, se usa una TU para los procesos de transformación y cuantificación. Una CU dada que tiene una o más PU también puede incluir una o más TU. Tras la predicción, el codificador de vídeo 20 puede calcular los valores residuales correspondientes a la PU. Los valores residuales comprenden valores de diferencias de

píxeles que se pueden transformar en coeficientes de transformación, cuantificar y escanear mediante las TU, para generar coeficientes de transformación en serie para la codificación por entropía. Esta divulgación usa típicamente el término "bloque de vídeo", o simplemente "bloque", 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 arbolado, es decir, una LCU o una CU, que incluye un nodo de codificación y unas PU y TU.

[0048] 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, en una cabecera de una o más de las imágenes o en otras ubicaciones, que describen un cierto número 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. Un codificador de vídeo 20 actúa típicamente sobre bloques de vídeo dentro de fragmentos de vídeo individuales con el 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 presentar tamaños fijos o variables y pueden diferir en tamaño de acuerdo con una norma de codificación especificada.

[0049] En un ejemplo, el HM soporta la predicción en diversos tamaños de PU. Suponiendo que el tamaño de una CU particular es $2N \times 2N$, el HM admite predicción intra en tamaños de PU de $2N \times 2N$ o $N \times N$ e predicción inter en tamaños de PU simétricos de $2N \times 2N$, $2N \times N$, $N \times 2N$ o $N \times N$. El HM también admite la división asimétrica para la predicción inter 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 en el 25 % y el 75 %. La parte de la CU correspondiente a la división del 25 % está indicada por una "n" seguida de una indicación de "arriba", "abajo", "izquierda" o "derecha". Así pues, por ejemplo, " $2N \times nU$ " se refiere a una CU de tamaño $2N \times 2N$ que está dividida horizontalmente con una PU de tamaño $2N \times 0,5N$ encima y una PU de tamaño $2N \times 1,5N$ debajo.

[0050] En esta divulgación, " $N \times N$ " y "N por N" pueden usarse indistintamente para hacer referencia a las dimensiones de píxeles de un bloque de vídeo en términos de dimensiones verticales y horizontales, por ejemplo, 16×16 píxeles o 16 por 16 píxeles. En general, un bloque de tamaño 16×16 tendrá 16 píxeles en la dirección vertical ($y = 16$) y 16 píxeles en la dirección horizontal ($x = 16$). Asimismo, un bloque de tamaño $N \times N$ presenta, 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 en un bloque pueden disponerse en filas y columnas. Además, no es necesario que los bloques presenten necesariamente el mismo número de píxeles en la dirección horizontal y en la dirección vertical. Por ejemplo, los bloques pueden comprender $N \times M$ píxeles, donde M no es necesariamente igual a N.

[0051] Tras la codificación de predicción intra o predicción inter 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 de píxeles) y las TU pueden comprender coeficientes en el dominio de las transformaciones tras la aplicación de una transformación, por ejemplo, una transformación de coseno discreta (DCT), una transformación de enteros, una transformación wavelet o una transformación similar desde un punto de visualización conceptual a los datos de vídeo residuales. Los datos residuales pueden corresponder a diferencias de píxeles entre píxeles de la imagen no codificada y los 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 generar coeficientes de transformación para la CU.

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

[0053] En algunos ejemplos, el codificador de vídeo 20 puede usar un escaneado predefinido, o un orden de "escaneado" para escanear los coeficientes de transformación cuantificados, para producir un vector en serie que se pueda codificar por entropía. En otros ejemplos, el codificador de vídeo 20 puede realizar un escaneado adaptable. Después de escanear los coeficientes de transformación 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 adaptable al contexto (CAVLC), la codificación aritmética binaria adaptable al contexto (CABAC), la codificación aritmética binaria adaptable 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.

[0054] Para realizar la CABAC, el codificador de vídeo 20 puede asignar un contexto dentro de un modelo contextual a un símbolo que se va a transmitir. El contexto puede referirse, por ejemplo, a si los valores contiguos del símbolo son iguales a cero. Para realizar la CAVLC, el codificador de vídeo 20 puede seleccionar un código de

longitud variable para un símbolo que se va a transmitir. Las palabras de código en VLC pueden construirse de forma que los códigos relativamente más cortos correspondan a símbolos más probables, mientras que los códigos relativamente más largos correspondan a símbolos menos probables. De esta manera, 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 puede basarse en un contexto asignado al símbolo.

[0055] En algunos ejemplos, las técnicas de esta divulgación están dirigidas al acceso aleatorio en codificación de vídeo. En particular, esta divulgación describe varias técnicas para codificar secuencias de vídeo que incluyen una o más tramas, o imágenes, en el que una primera imagen codificada de una CVS particular en un flujo de bits conforme puede ser una imagen RAP que no es una imagen IDR. Por ejemplo, de acuerdo con las técnicas divulgadas, la primera imagen codificada puede ser una imagen de CRA.

[0056] En otras palabras, un flujo de bits que incluye una o más imágenes de una CVS, en el que una primera imagen codificada del flujo de bits es una imagen RAP no IDR, se puede considerar un flujo de bits "conforme" de acuerdo con las técnicas de esta divulgación. Dicho de otra manera, un decodificador de vídeo que se ajusta a las técnicas divulgadas puede decodificar dicho flujo de bits con éxito y de una manera predecible y definida. Específicamente, las técnicas de esta divulgación incluyen procedimientos de manipulación, mediante el decodificador de vídeo, de la decodificación, así como las propiedades de salida y referencia, de imágenes principales asociadas con la primera imagen codificada. De forma alternativa, las técnicas también incluyen generar, mediante un codificador de vídeo, un flujo de bits conforme que excluye imágenes principales asociadas con una primera imagen codificada del flujo de bits que es una imagen RAP no IDR del flujo de bits, de modo que el flujo de bits pueda decodificarse con éxito mediante un decodificador de vídeo de una manera predecible y definida.

[0057] En esta divulgación, una imagen IDR de una CVS puede referirse en general a una imagen incluida dentro de la CVS que se codifica usando codificación intra-predictiva, es decir, una imagen "I" codificada sin referirse a ninguna otra imagen dentro o fuera de la CVS. Además, la imagen IDR puede referirse a una imagen para la cual todas las demás imágenes incluidas dentro de la CVS después de la imagen IDR de acuerdo con un orden de decodificación asociado con la CVS son decodificadas sin referencia a ninguna imagen que preceda a la imagen IDR de acuerdo con el orden de decodificación. Por ejemplo, de acuerdo con algunas técnicas (por ejemplo, H.264/MPEG-4 Parte 10/AVC, en adelante "H.264/AVC"), una CVS puede incluir una imagen IDR como una primera imagen de la CVS de acuerdo con una orden de decodificación asociada con la CVS, así como una o más imágenes IDR adicionales. Como un ejemplo, la CVS puede incluir uno o más GOP, en el que cada GOP comienza con una imagen IDR, seguida de una o más imágenes distintas de IDR (por ejemplo, las denominadas imágenes "P" y "B" codificadas usando la codificación inter-predictiva basada en la predicción hacia adelante y bidireccional de otras imágenes de referencia).

[0058] De acuerdo con las técnicas descritas anteriormente (por ejemplo, H.264/AVC), el acceso aleatorio de una CVS puede lograrse decodificando primero una imagen IDR de la CVS, por ejemplo, una imagen IDR de un GOP particular incluido dentro de la CVS. Debido a que las imágenes IDR pueden decodificarse sin referencia a ninguna otra imagen, como se describió anteriormente, de acuerdo con estas técnicas, el acceso aleatorio de la CVS puede realizarse sobre una base GOP decodificando primero una imagen IDR localizada al comienzo de cada GOP. En otras palabras, de acuerdo con algunas técnicas (por ejemplo, H.264/AVC), el acceso aleatorio de una CVS puede realizarse solo a partir de una imagen IDR incluida dentro de la CVS. Como tal, en estas técnicas, para que una primera imagen codificada de una CVS particular en un flujo de bits conforme sea una imagen RAP, la imagen debe ser una imagen IDR.

[0059] En contraste con las técnicas descritas anteriormente, de acuerdo con las técnicas de esta divulgación, el acceso aleatorio de un flujo de bits a partir de una imagen no IDR (por ejemplo, una imagen de CRA) puede realizarse de forma predecible y definida, o "estándar" mediante la adaptación de decodificadores de vídeo. Como resultado, las técnicas divulgadas pueden mejorar significativamente la interoperabilidad de los sistemas y dispositivos codificadores de vídeo y decodificadores de vídeo, así como la experiencia del usuario, en general, para el acceso aleatorio de flujo de bits que puede ocurrir frecuentemente en diversas aplicaciones de vídeo. Por ejemplo, las técnicas descritas en el presente documento pueden incluir al menos uno o más de los siguientes aspectos novedosos, en comparación con otras técnicas:

(1) detectar cuándo se produce un acceso aleatorio desde una imagen RAP no IDR (por ejemplo, una imagen de CRA);

(2) identificar y decodificar una o más imágenes que siguen a la imagen RAP no IDR en un orden de decodificación, pero preceden a la imagen RAP no IDR en un orden de salida (es decir, una o más "imágenes principales" de la imagen RAP no IDR); y

(3) especificar que cada una de las imágenes principales de la imagen RAP no IDR no se emite, incluso en el caso de que un `output_flag` señalado correspondiente sea verdadero, o "1" (es decir, el indicador de salida indica que la respectiva imagen se va a mostrar), y que la imagen respectiva no se utiliza como una imagen de

referencia para ninguna otra imagen que siga la imagen RAP no IDR en el orden de descodificación y el orden de salida.

5 **[0060]** Como se describió anteriormente, de acuerdo con algunas técnicas (por ejemplo, H.264/AVC), una imagen IDR puede servir como un punto de acceso convencional (por ejemplo, un punto de acceso aleatorio, o imagen "RAP") para una CVS. Por ejemplo, la imagen IDR puede incluirse al comienzo de una parte descodificable independientemente de la CVS, a veces denominada GOP. Esta implementación del acceso aleatorio de una CVS a veces se denomina implementación de "GOP cerrado", en el que ninguna imagen dentro de un GOP particular se refiere a ninguna imagen que ocurra antes de una imagen IDR del GOP, por ejemplo, imágenes incluidas en un
10 GOP anterior de la CVS, o un GOP de otra, CVS precedente, de acuerdo con un orden de descodificación asociado con la CVS. Como ya se explicó anteriormente, en este contexto, un GOP se puede definir como una imagen IDR seguida de una o más imágenes "P" y/o "B".

15 **[0061]** En una implementación llamada "GOP abierto", una imagen de CRA tiene un propósito similar al de la imagen IDR descrita anteriormente con referencia a la implementación de GOP cerrada. Por ejemplo, en este contexto, un GOP puede definirse como una imagen de CRA seguida de una o más imágenes "P" y/o "B". Sin embargo, en contraste con la implementación cerrada del GOP, en la implementación GOP abierta, las imágenes incluidas en un GOP particular pueden referirse a imágenes que ocurren antes de una imagen de CRA del GOP, por ejemplo, imágenes incluidas en un GOP anterior de la CVS, o un GOP de otro, que precede a CVS, de acuerdo
20 con un orden de descodificación asociado con la CVS. Por ejemplo, de forma coherente con la implementación abierta del GOP, una imagen "B" que sigue a una imagen de CRA (que, como la imagen IDR, es una imagen intra-predicha, o "I") de un GOP de una CVS de acuerdo con un orden de descodificación asociado con la CVS puede referirse a una imagen (por ejemplo, una imagen "P" o "B") incluida dentro de un GOP anterior de la CVS.

25 **[0062]** De acuerdo con algunas técnicas, una imagen "B" de una CVS se predice convencionalmente haciendo referencia a una imagen que precede a la imagen "B" y una imagen que sigue a la imagen "B" en un orden de salida asociado con la CVS. Por ejemplo, la imagen "B" de este ejemplo puede referirse (es decir, usar como imagen de referencia) a la imagen incluida en el GOP anterior, que puede preceder a la imagen "B" en un orden de salida asociado con la CVS, y también referirse a (es decir, usar como imagen de referencia) la imagen de CRA, que puede seguir a la imagen "B" en el orden de salida. En otras palabras, en este ejemplo, la imagen "B" sigue a
30 la imagen de CRA en el orden de descodificación, pero precede a la imagen de CRA en el orden de salida. Como tal, la imagen "B" puede considerarse una "imagen principal" de la imagen de CRA. Sin embargo, en otros ejemplos, la imagen "B" puede ser cualquier otro tipo de imagen que también sea una imagen principal de la imagen de CRA, como se definió anteriormente.

35 **[0063]** El ejemplo descrito anteriormente ilustra al menos un problema asociado con la implementación de GOP abierta descrita anteriormente. Específicamente, en los casos en que el acceso aleatorio de una CVS se realiza desde una imagen de CRA incluida dentro de la CVS, las imágenes principales de la imagen de CRA no se pueden descodificar correctamente. Esto se debe al hecho de que, en los casos en que la imagen de CRA es la primera
40 imagen codificada de la CVS, las imágenes que preceden a la imagen de CRA en un orden de descodificación asociado con la CVS no se descodifican y, por lo tanto, no están disponibles como referencia imagen de las imágenes principales. En consecuencia, en la implementación de GOP abierta descrita anteriormente, las imágenes principales no pueden descodificarse correctamente, y por lo tanto pueden perjudicar la experiencia del usuario si se muestran. Por ejemplo, si se descodifica, las imágenes principales pueden incluir datos de vídeo
45 erróneos y, si se muestran, pueden degradar la calidad visual de las imágenes mismas, así como de la CVS en general. Por las mismas razones, en la implementación GOP abierta, otras imágenes de la CVS que siguen a la imagen de CRA tanto en orden de descodificación como en orden de salida (p. ej., imágenes "P") pueden no referirse a las imágenes principales (por ejemplo, ya que las imágenes principales, si se descodifican, pueden incluir datos de vídeo erróneos), o cualquier otra imagen que preceda a la imagen de CRA en el orden de
50 descodificación y el orden de salida (por ejemplo, ya que estas imágenes no están descodificadas, y por lo tanto no están disponibles como imágenes de referencia).

55 **[0064]** Hablando en términos generales, cualquiera de las técnicas descritas anteriormente (es decir, la implementación de GOP cerrada que usa imágenes IDR y la implementación de GOP abierta que usa imágenes de CRA) puede permitir el acceso aleatorio de una CVS de datos de vídeo. Sin embargo, de acuerdo con algunas normas de codificación, tales como, por ejemplo, H.264/AVC, un flujo de bits que comienza con una imagen de CRA se considera un flujo de bits "no conforme". Por ejemplo, como se describió anteriormente, de acuerdo con algunas técnicas, tales como, por ejemplo, H.264/AVC, un flujo de bits debe comenzar con una imagen IDR. En
60 otras palabras, de acuerdo con estas técnicas, solo la implementación de acceso aleatorio de GOP cerrada descrita anteriormente puede soportarse. Las técnicas de esta divulgación pueden permitir que un descodificador de vídeo maneje tal flujo de bits no conforme (es decir, un flujo de bits que comienza con una imagen de CRA y se ajusta a la implementación de GOP abierta). Dicho de otra manera, las técnicas descritas en el presente documento pretenden definir dicho flujo de bits como un flujo de bits "conforme". En algunos ejemplos, un flujo de bits conforme a las técnicas de esta divulgación incluye flujos de bits que comienzan con imágenes de CRA y se ajustan a la
65 implementación de GOP abierto, así como flujos de bits que comienzan con imágenes de IDR y se ajustan a la implementación de GOP cerrada.

[0065] Como ya se explicó anteriormente, un problema identificado con respecto al acceso aleatorio que ocurre en una imagen de CRA se relaciona con el hecho de que las imágenes principales de la imagen de CRA pueden no descodificarse correctamente y, por lo tanto, pueden perjudicar la experiencia del usuario si se muestran. Las técnicas de esta divulgación pueden abordar este problema habilitando el acceso aleatorio de una CVS desde una imagen de CRA realizando la descodificación, así como manipulando las propiedades de salida y referencia, de imágenes principales asociadas con la imagen de CRA de una manera particular. Específicamente, las técnicas pueden incluir algunos o todos de los siguientes pasos:

Paso 1: Identificar una o más imágenes de una CVS como imágenes principales de una imagen de CRA de la CVS cuando un valor de recuento de orden de imagen (POC) de cada una de las imágenes es menor que un valor POC de la imagen de CRA (es decir, el respectivo la imagen precede a la imagen de CRA en un orden de salida asociado con la CVS), y cuando la imagen respectiva sigue a la imagen de CRA en un orden de descodificación asociado con la CVS.

Paso 2: Determinar, para cada una de las imágenes principales, si la imagen principal respectiva hace referencia a una imagen que no está disponible para descodificar.

Paso 3: Generar, para cada una de las una o más imágenes principales que se determina que hacen referencia a una imagen que no está disponible para descodificar una imagen de referencia virtual (por ejemplo, generar una imagen de luma (o croma) "media" que tiene valores de luma (o croma) que corresponden a un valor medio de un rango de valores de luma (o croma) asociado con la CVS, por ejemplo, una imagen "gris").

Paso 4: Descodificar cada una de las una o más imágenes principales para las que se genera una imagen de referencia virtual utilizando la imagen de referencia virtual generada correspondiente, así como descodificar las imágenes principales restantes. (La descodificación de una o más imágenes principales se realiza para mantener los parámetros de temporización CVS originales en el descodificador de vídeo, por ejemplo, dentro de un DPB del descodificador de vídeo, aunque, como se muestra a continuación, las imágenes principales descodificadas no se pueden emitir o utilizar como imágenes de referencia para otras imágenes de la CVS).

Paso 5: Establecer un output_flag asociado con cada una de las una o más imágenes principales descodificadas o falsa, o "0", para no generar la imagen principal respectiva, incluso en el caso de que un indicador de salida actual sea igual a verdadero, o "1". (De forma alternativa, las técnicas pueden incluir simplemente ignorar, o "enmascarar" el output_flag actual que es igual a verdadero, o "1", para no emitir la imagen principal respectiva).

Paso 6: Evitar que cada una de las una o más imágenes principales descodificadas se use como una imagen de predicción (es decir, una referencia) para cualquier otra imagen de la CVS que siga a la imagen de CRA tanto en el orden de descodificación como en el orden de salida.

[0066] Además, las técnicas descritas en el presente documento pueden aplicarse a una codificación (por ejemplo, un codificador de vídeo 20), en lugar de a un dispositivo de descodificación (por ejemplo, descodificador de vídeo 30). Por ejemplo, en casos donde una primera imagen codificada de una CVS comprende una imagen de CRA, un codificador de vídeo inteligente que se ajusta a las técnicas de esta divulgación puede configurarse para evitar el envío de cualquier imagen principal de la imagen de CRA a un descodificador de vídeo. Como un ejemplo, el codificador de vídeo puede configurarse para enviar solo imágenes "P" que siguen a la imagen de CRA de acuerdo con un orden de descodificación asociado con la CVS. Para lograr esto, el codificador de vídeo puede configurarse para generar un denominado "subconjunto" de flujo de bits al eliminar todas las "unidades de acceso" o conjuntos comparables de datos que contienen imágenes principales asociadas con la imagen de CRA, en algunos ejemplos. En consecuencia, en el ejemplo alternativo ilustrado anteriormente, un codificador de vídeo, en lugar de un descodificador de vídeo, puede configurarse para manejar (es decir, eliminar) las imágenes principales de un CRA de una CVS como parte de generar un flujo de bits que incluye la CVS, para mejorar la interoperabilidad y la experiencia del usuario para el acceso aleatorio del flujo de bits en el descodificador de vídeo.

[0067] Como tal, de acuerdo con las técnicas descritas en el presente documento, una primera imagen codificada de una CVS de acuerdo con un orden de descodificación asociado con la CVS en un flujo de bits conforme puede ser una imagen IDR o una imagen de CRA. En otras palabras, las técnicas de esta divulgación pueden permitir el acceso aleatorio que se produce en una imagen de CRA de una CVS definiendo un flujo de bits en el que una primera imagen codificada de una CVS de acuerdo con un orden de descodificación asociado con la CVS es una imagen de CRA como un flujo de bits conforme. Por ejemplo, las técnicas de esta divulgación pueden ser aplicables a una norma de codificación particular (por ejemplo, H.265/HEVC), o una extensión de una norma de codificación (por ejemplo, H.264/AVC). En cualquier caso, de acuerdo con las técnicas descritas, dicho flujo de bits puede ser un flujo de bits conforme. En otras palabras, dicho flujo de bits puede descodificarse con éxito mediante un descodificador de vídeo que se ajuste a las técnicas de esta divulgación de una manera definida y predecible.

[0068] La siguiente descripción proporciona información adicional y ejemplos relacionados con las técnicas de esta divulgación descritas anteriormente, así como también información y técnicas adicionales.

[0069] Específicamente, las técnicas descritas en el presente documento pueden incluir uno o más de los siguientes aspectos novedosos, en comparación con otras técnicas: (1) detectar cuándo se produce un acceso aleatorio desde una imagen no IDR; (2) especificar que una imagen no se emite, incluso en el caso de que un indicador de salida señalado correspondiente para la imagen sea igual a verdadero, o "1;" y (3) señalización de parámetros de momento de eliminación de "memoria intermedia codificada" (CPB) para imágenes que siguen a una imagen RAP no IDR en un orden de decodificación, cuando la imagen RAP no IDR es una primera imagen codificada de un flujo de bits, y cuando las imágenes asociadas con la primera imagen codificada no están presentes. En algunos ejemplos coherentes con las técnicas divulgadas, los parámetros de momento de eliminación de CPB actualizados pueden indicarse mediante un desplazamiento que se aplica a todas las imágenes siguientes a la imagen RAP no IDR en el orden de decodificación después de realizar un acceso aleatorio desde la imagen RAP no IDR.

[0070] Las técnicas descritas en el presente documento pueden ser aplicables a varias normas de codificación de vídeo, incluyendo 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 ajustable a escala (SVC) y de codificación de vídeo de múltiples visualizaciones (MVC). Además, las técnicas divulgadas pueden ser aplicables a la norma HEVC actualmente desarrollada por el JCT-VC del Grupo de expertos en codificación de vídeo (VCEG) ITU-T y el Grupo de expertos en imágenes en movimiento (MPEG) ISO/IEC. Como se explicó anteriormente, una versión particular de HEVC a la que se hace referencia en esta divulgación es WD4 descrita en el documento JCTVC-F803.

[0071] Ahora se describirán algunas técnicas de gestión DPB. De acuerdo con algunas técnicas de codificación de vídeo, pueden implementarse diversos procedimientos de gestión de DPB. Como un ejemplo, las imágenes decodificadas usadas para predecir imágenes codificadas subsiguientes, y para salidas futuras, pueden almacenarse en un DPB. Para utilizar de manera eficiente la memoria de un DPB, se pueden especificar procesos de gestión de DPB, que incluyen un proceso de almacenamiento de imágenes decodificadas en el DPB, un proceso de marcado de imágenes de referencia y procesos de salida y eliminación de imágenes decodificadas del DPB. La gestión de DPB puede incluir al menos los siguientes aspectos: (1) identificación de la imagen e identificación de la imagen de referencia; (2) construcción de lista de imágenes de referencia; (3) marcado de imágenes de referencia; (4) salida de imagen desde el DPB; (5) inserción de imagen en el DPB; y (6) eliminación de imagen del DPB. A continuación se proporciona alguna introducción al marcado de imágenes de referencia y la construcción de la lista de imágenes de referencia.

[0072] Como un ejemplo, ahora se describirán las técnicas de marcado de la lista de imágenes de referencia. De acuerdo con algunas técnicas de codificación de vídeo, pueden implementarse diversos procedimientos de marcado de imágenes de referencia. Como un ejemplo, el marcado de imágenes de referencia en H.264/AVC se puede resumir de la siguiente manera. Un número máximo, que puede denominarse "M" (por ejemplo, correspondiente al elemento sintáctico `num_ref_frames`), de imágenes de referencia utilizadas para la interproducción se puede indicar en un conjunto de parámetros de secuencia activa (SPS). Cuando se decodifica una imagen de referencia, se puede marcar como "utilizada como referencia". Si la decodificación de la imagen de referencia hace que se marquen más de "M" imágenes como "utilizadas como referencia", al menos una imagen puede marcarse como "no utilizada para referencia". Posteriormente, el proceso de eliminación de DPB puede eliminar las imágenes marcadas como "no utilizadas para referencia" de la DPB, si las imágenes tampoco son necesarias para la salida.

[0073] Cuando se decodifica una imagen, puede ser una imagen que no sea de referencia o una imagen de referencia. Una imagen de referencia puede ser una imagen de referencia a largo plazo, o una imagen de referencia a corto plazo, y, cuando se marca como "no utilizada para referencia", la imagen puede convertirse en una imagen sin referencia.

[0074] H.264/AVC incluye operaciones de marcado de imágenes de referencia que cambian el estado de las imágenes de referencia. Por ejemplo, en H.264/AVC, hay dos tipos de operaciones para el marcado de imágenes de referencia, a saber, la ventana deslizante y el control de memoria adaptable. El modo de funcionamiento para el marcado de imágenes de referencia se selecciona basándose en la imagen. Como un ejemplo, el marcado de imágenes de referencia de ventana deslizante funciona como una cola de primero en entrar, primero en salir (FIFO) con un número fijo de imágenes de referencia a corto plazo. En otras palabras, primero se debe eliminar una imagen de referencia a corto plazo con un tiempo de decodificación anterior (es decir, marcado como una imagen "no utilizada como referencia"), de manera implícita. Como otro ejemplo, el marcado de imágenes de referencia de control de memoria adaptable elimina explícitamente las imágenes a corto o largo plazo. También permite cambiar el estado de las imágenes a corto y largo plazo.

[0075] Como otro ejemplo, ahora se describirán las técnicas de construcción de la lista de imágenes de referencia. De acuerdo con algunas técnicas de codificación de vídeo, pueden implementarse diversos

procedimientos de construcción de lista de imágenes de referencia. Como un ejemplo, típicamente, una construcción de lista de imágenes de referencia para una primera o una segunda lista de imágenes de referencia de una imagen "B" puede incluir dos pasos: (1) inicialización de la lista de imágenes de referencia, y (2) reordenación de la lista de imágenes de referencia (que puede denominarse "modificación"). La inicialización de la lista de imágenes de referencia puede ser un mecanismo explícito que coloca imágenes de referencia en una memoria de imágenes de referencia (también conocida como DPB) en una lista basándose en un orden de valores de POC (que, como se explicó anteriormente, es un "Recuento de orden de imagen" y está alineado con un orden de salida, o un orden de visualización, de una imagen).

[0076] El mecanismo de reordenación de la lista de imágenes de referencia puede modificar una posición de una imagen introducida en la lista durante la inicialización de la lista de imágenes de referencia en cualquier posición nueva, o colocar cualquier imagen de referencia en la memoria de imágenes de referencia en cualquier posición, incluso si la imagen no pertenece a la lista inicializada. Algunas imágenes, después del reordenamiento (o modificación) de la lista de imágenes de referencia, pueden colocarse en posiciones muy "lejanas" en la lista. Sin embargo, si una posición de una imagen excede un número de imágenes de referencia activas de la lista, la imagen puede no ser considerada como una entrada de la lista de imágenes de referencia final. El número de imágenes de referencia activas se puede señalar en una cabecera de fragmento para cada lista.

[0077] De forma alternativa, se ha descrito un enfoque diferente para la gestión de DPB en el documento "JCTVC-F493: Absolute Signaling of Reference Pictures" ["JCTVC-F493: Señalización absoluta de imágenes de referencia"] por Sjöberg et al., 6.^a reunión, Torino, 2011 (denominada en adelante JCTVC-F493), cuyo contenido completo se incorpora en el presente documento como referencia.

[0078] Ahora se describirán algunas técnicas de conjuntos de imágenes de referencia (RPS). Por ejemplo, la solicitud de patente de Estados Unidos n.º 13/622,972, presentada el 19 de septiembre de 2012, cuyo contenido completo también se incorpora en el presente documento como referencia, describe un RPS, que incluye, para cada imagen, una serie de imágenes de referencia que pueden ser utilizadas por la imagen actual, o actualmente codificada, y una imagen que sigue a la imagen codificada actualmente en un orden de descodificación. Se puede proporcionar una definición detallada de un RPS de la siguiente manera: un conjunto de imágenes de referencia asociadas con una imagen, que consiste en todas las imágenes de referencia, excluyendo la imagen asociada, que pueden usarse para la predicción inter de la imagen asociada o cualquier imagen siguiente la imagen asociada en un orden de descodificación, y que tiene el elemento sintáctico temporal_id menor o igual que el de la imagen asociada.

[0079] Se describirán ahora ejemplos de RAP y un RPS correspondiente. Como se explicó previamente, en esta divulgación, "acceso aleatorio" se refiere a una descodificación de una CVS, comenzando desde una imagen codificada que no es una primera imagen codificada tradicional, es decir, una imagen IDR, en la CVS. Una imagen RAP no IDR, que se puede denominar "picR", se puede definir como una imagen codificada para la que se cumplen todas las condiciones siguientes:

(1) picR no es una imagen IDR;

(2) deje que el POC de picR sea "rPoc" y deje que "picA" sea una imagen en el mismo CVS y siguiente a picR tanto en orden de descodificación como en orden de salida, y permita que el POC de picA sea "aPoc". Cuando se realiza un acceso aleatorio en picR, todas las imágenes que están en el mismo CVS y siguen a picA en el orden de salida se pueden descodificar correctamente.

[0080] En este ejemplo, para una imagen RAP no IDR picR, si la siguiente condición es verdadera, la imagen puede denominarse imagen de CRA: cuando se realiza un acceso aleatorio en picR, todas las imágenes que están en el mismo CVS y siguen picR en el orden de salida se puede descodificar correctamente. Si la condición anterior no es verdadera para una imagen RPR no IDR, la imagen puede denominarse una imagen de actualización gradual de descodificación (GDR). Además, para una imagen de CRA, un RPS correspondiente puede no contener ninguna imagen de referencia para la imagen de CRA, pero típicamente puede contener al menos una imagen para las imágenes que siguen a la imagen de CRA en el orden de descodificación.

[0081] La FIG. 4 es un diagrama conceptual que ilustra un ejemplo de jerarquías de referencia entre imágenes de GOP de datos de vídeo, coherente con las técnicas de esta divulgación. En particular, la FIG. 4 ilustra una codificación de imagen "B" jerárquica con cuatro niveles temporales y un tamaño GOP de "8". Como se muestra en la FIG. 4, cuando una imagen con un valor POC igual a "8" se codifica como intra- (es decir, una imagen "I"), la imagen puede ser una imagen de CRA. Basándose en la definición del RPS, el RPS contiene una imagen con un valor de POC igual a "0" para las imágenes que siguen a esta imagen en el orden de descodificación.

[0082] Ahora se describirán las imágenes principales y los RPS correspondientes. Como se explicó anteriormente, las imágenes que siguen a una imagen RAP en un orden de descodificación, pero que preceden a la imagen RAP en un orden de visualización, pueden denominarse "imágenes principales" correspondientes de la

imagen RAP. En el ejemplo de la FIG. 4, los RPS de las imágenes principales correspondientes de una imagen de CRA (es decir, la imagen con un valor POC de "8") son como se muestra en la Tabla I a continuación.

Tabla I

5

POC	RPS
4	{0, 8}
2	{0, 8, 4}
1	{0, 8, 4, 2}
3	{8, 4, 2}
6	{8, 4, 2}
5	{8, 4, 6}
7	{8, 6}

10 **[0083]** Ahora se describirán las técnicas de salida de la imagen. En HEVC WD4, a cada imagen se le puede asignar un output_flag. Cuando este indicador es igual a falso, o "0", la imagen respectiva no se utiliza para la salida y, por lo tanto, no se mostrará.

15 **[0084]** Se describirá ahora un ejemplo de una CPB. Es posible que se necesite una CPB mediante un decodificador de vídeo para la recepción y almacenamiento en memoria intermedia de las unidades de acceso, cada una de las cuales contiene una imagen codificada y unidades de capa de abstracción de red (NAL) asociadas, antes de descodificarlas. En HEVC WD4, por ejemplo, faltan las operaciones de CPB, pero las operaciones de CPB como se especifica en H.264/AVC pueden, no obstante, aplicarse. Como un ejemplo, de acuerdo con H.264/AVC, para un flujo de bits conforme, se puede cumplir en su totalidad una lista de condiciones tal como se especifica en la subcláusula C.3 en H.264/AVC. Dos de las condiciones de conformidad de flujo de bits son las siguientes:

20

(1) Un desbordamiento de CPB se especifica como una condición en la que un número total de bits en una CPB es mayor que un tamaño de CPB. La CPB nunca puede desbordarse;

25

(2) Un subdesbordamiento de CPB se especifica como una condición en la que $t_{r,n}(n)$ es menor que " $t_{af}(n)$ ". Cuando low_delay_hrd_flag es igual a "0", es posible que la CPB nunca se subdesborde.

30 **[0085]** Ahora se analizarán algunos posibles problemas con las técnicas descritas anteriormente. Los diversos enfoques descritos anteriormente, relacionados con el acceso aleatorio que se produce en una imagen de CRA, tienen varios inconvenientes. Como ejemplo, en el caso de que una imagen principal no esté presente, un decodificador conforme puede no saber si la imagen principal se pierde debido a pérdidas de transmisión, o debido a descartes de imagen intencionales (por ejemplo, por un servidor de transmisión, intencionalmente para el funcionamiento de acceso aleatorio). Como otro ejemplo, las imágenes principales pueden no descodificarse correctamente y, por lo tanto, pueden perjudicar la experiencia del usuario si se muestran. Como otro ejemplo más, en el caso de que algunas imágenes codificadas caigan intencionalmente, el flujo de bits resultante comenzando desde una imagen de CRA puede entrar en conflicto con una o más condiciones de conformidad de flujo de bits, y, por lo tanto, puede producirse un comportamiento de descodificación no previsible y resultados de descodificación incontrolables. Por ejemplo, cuando se descartan todas las imágenes principales, una imagen siguiente después de la imagen de CRA en un orden de descodificación se convierte en una imagen codificada después de la última imagen principal en el orden de descodificación. En comparación con el caso en el que están presentes las imágenes principales, esta imagen siguiente fluye hacia una CPB más temprano, y, en consecuencia, su tiempo de llegada final de CPB $t_{af}(n)$ se vuelve anterior. El momento de eliminación de CPB obtenido del elemento sintáctico cpb_removal_delay en el mensaje de información de mejora suplementaria (SEI) de temporización de imagen asociado puede no cambiar. Por lo tanto, no se producirá subdesbordamiento de CPB. Sin embargo, si un número de bits de esta imagen y las siguientes imágenes en el orden de descodificación es significativamente mayor que el de las imágenes principales descartadas, puede producirse un desbordamiento de CPB.

45 **[0086]** Esta divulgación describe varias técnicas que pueden, en algunos casos, reducir o eliminar algunos de los inconvenientes descritos anteriormente. En particular, las técnicas de esta divulgación pueden resolver los problemas descritos anteriormente, relacionados con el acceso aleatorio de imágenes de CRA, empleando diversos procedimientos para asegurar que un flujo de bits para el que una primera imagen codificada es una

50

imagen de CRA sea conforme, independientemente de si las imágenes principales asociadas con la imagen de CRA están presentes. Las técnicas divulgadas incluyen al menos los siguientes aspectos que pueden usarse para implementar las características descritas anteriormente.

5 **[0087]** Como ejemplo, se puede agregar opcionalmente un proceso para la detección de cuándo se produce un acceso aleatorio. La detección de cuándo se produce el acceso aleatorio para cada imagen y para cada imagen principal, ya sea que esté dirigida a la descodificación y/o salida, puede realizarse mediante la construcción de un conjunto de imágenes desaparecidas (VPS) que contiene imágenes que pueden no recibirse ni descodificarse correctamente debido al acceso aleatorio, pero se recibirán y descodificarán correctamente en un caso normal.
10 Siempre que el VPS no esté vacío, la detección puede ser necesaria.

[0088] Como otro ejemplo, se puede realizar el manejo de una propiedad de salida de imágenes principales, de modo que las imágenes principales no se pueden usar para salida cuando se produce el acceso aleatorio, comenzando desde una imagen de CRA asociada.

15 **[0089]** Como otro ejemplo, los procesos de descodificación para una imagen principal pueden modificarse, de modo que, si se recibe una imagen principal, solo el análisis de sintaxis de alto nivel y la invocación de procesos de descodificación asociados, por ejemplo, obtención de un conjunto de imágenes de referencia (RPS), pueden realizarse, y se puede omitir la descodificación de dicha imagen.

20 **[0090]** Como otro ejemplo, se puede agregar una restricción de flujo de bits, de modo que, incluso cuando un descodificador comienza a descodificar una imagen de CRA, y no hay imágenes principales después de la imagen de CRA, información relacionada con la conformidad de CPB, incluyendo parámetros de descodificador hipotético de referencia (HRD) mensajes SEI de período de almacenamiento de imágenes en memoria intermedia, y mensajes SEI de temporización de imagen, pueden usarse para cumplir con las restricciones de CPB, y, por lo tanto, no puede producirse desbordamiento ni subdesbordamiento de la memoria intermedia.

25 **[0091]** Como otro ejemplo, un mensaje SEI asociado con una imagen de CRA puede señalarse para incluir un conjunto adicional de parámetros de retardo inicial de CPB, de modo que, cuando las imágenes principales no están presentes, las restricciones de conformidad CPB pueden cumplirse cuando se aplica el conjunto adicional de parámetros de retardo inicial de CPB. Más específicamente, en el mensaje SEI del período de almacenamiento en memoria intermedia de imágenes, se pueden señalar dos conjuntos de parámetros de retardo inicial de CPB si la imagen actual es una imagen de CRA.

30 **[0092]** Los siguientes ejemplos demuestran las características descritas anteriormente de las técnicas de esta divulgación. Para describir los siguientes ejemplos, se definen los siguientes términos:

35 **imagen principal:** Una imagen, asociada a una imagen de CRA, que sucede a la imagen de CRA en un orden de descodificación y precede a la imagen de CRA en orden de salida o visualización.

40 **VPS:** Un conjunto de imágenes de referencia, asociadas con una imagen de CRA, que tienen un orden de visualización anterior al de la imagen de CRA.

45 **[0093]** Se describirán ahora ejemplos de sintaxis y, en particular, ejemplos de sintaxis de mensaje SEI del período de almacenamiento en memoria intermedia.

TABLA II

	C	Descriptor
buffering_period(payloadSize) {		
seq_parameter_set_id	5	ue(v)
cra_para_present_flag	5	u(l)
if(NalHrdBpPresentFlag) {		
for(SchedSelIdx = 0; SchedSelIdx <= cpb_cnt_minus1; SchedSelIdx++) {		
initial_cpb_removal_delay[SchedSelIdx]	5	u(v)
initial_cpb_removal_delay_offset[SchedSelIdx]	5	u(v)
if (cra_para_present_flag) {		
update_initial_cpb_removal_delay[SchedSelIdx]	5	u(v)
update_initial_cpb_removal_delay_offset[SchedSelIdx]	5	u(v)
}		
}		
}		
if(VclHrdBpPresentFlag) {		
for(SchedSelIdx = 0; SchedSelIdx <= cpb_cnt_minus1; SchedSelIdx++) {		
initial_cpb_removal_delay[SchedSelIdx]	5	u(v)
initial_cpb_removal_delay_offset[SchedSelIdx]	5	u(v)
if (cra_para_present_flag) {		
update_initial_cpb_removal_delay[SchedSelIdx]	5	u(v)
update_initial_cpb_removal_delay_offset[SchedSelIdx]	5	u(v)
}		
}		
}		
}		

5 **[0094]** Además, a continuación se describen ejemplos de la semántica del mensaje SEI del período de almacenamiento en memoria intermedia:

10 **[0095]** Como un ejemplo, cra_para_present_flag igual a verdadero, o "1", puede indicar que se señala otro conjunto de retardos iniciales de CPB cuando la imagen asociada actual es una imagen de CRA. Este indicador igual a falso, o "0", puede indicar que no se señala ningún conjunto adicional de retardos iniciales de CPB. Este indicador puede ser igual a "0" cuando la imagen asociada no es una imagen de CRA.

15 **[0096]** Como otro ejemplo, update_initial_cpb_removal_delay[SchedSelIdx] puede especificar el retardo para la SchedSelIdx-ésima CPB entre el momento de llegada en la CPB del primer bit de los datos codificados asociados a la unidad de acceso asociada al mensaje SEI período de almacenamiento en memoria intermedia y el momento de eliminación de la CPB de los datos codificados asociados a la misma unidad de acceso, para el primer período de almacenamiento en memoria intermedia después de la inicialización de HRD. El elemento sintáctico puede tener una longitud en bits dada por initial_cpb_removal_delay_length_minus1 + 1. Puede ser en unidades de un reloj de 90 kHz. Además, update_initial_cpb_removal_delay[SchedSelIdx] puede no ser igual a "0" y puede no exceder de $90000 * (CpbSize[SchedSelIdx] \div BitRate[SchedSelIdx])$, el tiempo equivalente del tamaño de la CPB en unidades de reloj de 90 kHz.

20 **[0097]** Como otro ejemplo, update_initial_cpb_removal_delay_offset[SchedSelIdx] puede ser utilizado para la SchedSelIdx-ésima CPB en combinación con el cpb_removal_delay para especificar el tiempo de entrega inicial

de las unidades de acceso codificado a la CPB. Por ejemplo, `update_initial_cpb_removal_delay_offset[SchedSelIdx]` puede estar en unidades de un reloj de 90 kHz. El elemento sintáctico `update_initial_cpb_removal_delay_offset[SchedSelIdx]` puede ser un código de longitud fija cuya longitud en bits viene dada por `initial_cpb_removal_delay_length_minus1 + 1`. Este elemento sintáctico puede no ser utilizado por los descodificadores y puede ser necesario solo para el programador de entrega (HSS) especificado en el Anexo C de HEVC WD4.

[0098] Se describirán ahora ejemplos de procesos de descodificación. De acuerdo con las técnicas de esta divulgación, los siguientes procesos de descodificación pueden añadirse y/o modificarse con relación a los procesos de descodificación descritos en la solicitud de patente de Estados Unidos n.º 13/622,972, presentada el 19 de septiembre de 2012.

[0099] Como un ejemplo, ahora se describirán las técnicas de creación de VPS. En algunos ejemplos, la creación de VPS puede ocurrir inmediatamente después de la invocación de un proceso de obtención para un RPS cuando una imagen actual es una imagen de CRA. Por ejemplo, en los casos en que la imagen actual es una imagen de CRA, si alguna imagen en el RPS no está en un DPB, el VPS se puede establecer como el RPS actual. De lo contrario, el VPS puede configurarse para estar vacío.

[0100] Como otro ejemplo, ahora se describirán las principales técnicas de identificación de imágenes. En algunos ejemplos, si un VPS no está vacío, y una imagen tiene un RPS que se superpone con el VPS, la imagen puede identificarse como una imagen principal.

[0101] Como otro ejemplo, se describirán ahora las técnicas de descodificación de imágenes principales. En algunos ejemplos, la descodificación de una imagen principal puede omitirse mediante un descodificador conforme.

[0102] Como otro ejemplo, ahora se describirán técnicas de salida de imágenes principales. En algunos ejemplos, para cada imagen principal, `output_flag` puede establecerse en "falso", independientemente de si el valor de `output_flag` en la cabecera de fragmento es igual a "0" o "1".

[0103] Ahora se describirá un HRD y, en particular, el funcionamiento de una CPB. Por ejemplo, las especificaciones en esta parte de la divulgación se pueden aplicar de forma independiente a cada conjunto de parámetros CPB que están presentes y a la conformidad tanto de Tipo I como de Tipo II.

[0104] Como un ejemplo, ahora se describirá el tiempo de llegada del flujo de bits. El HRD se puede inicializar en cualquiera de los mensajes SEI de período de almacenamiento en memoria intermedia. Antes de la inicialización, la CPB puede estar vacía. En este ejemplo, después de la inicialización, el HRD puede no inicializarse de nuevo mediante los posteriores mensajes SEI del período de almacenamiento en memoria intermedia. También en este ejemplo, si la primera unidad de acceso es una unidad de acceso CRA, y las imágenes principales no están presentes y `cra_para_present_flag` es igual a "1", `useUpdatePara` puede establecerse en "1". De lo contrario, `useUpdatePara` se puede establecer en "0".

[0105] Por ejemplo, si `useUpdatePara` es igual a "0", `InitialCpbRemovalDelay[SchedSelIdx]` se puede establecer en `initial_cpb_removal_delay[SchedSelIdx]` y `InitialCpbRemovalDelayOffset[SchedSelIdx]` se puede establecer en `initial_cpb_removal_delay_offset[SchedSelIdx]`. De lo contrario, `InitialCpbRemovalDelay[SchedSelIdx]` se puede establecer en `update_initial_cpb_removal_delay[SchedSelIdx]` e `InitialCpbRemovalDelayOffset[SchedSelIdx]` se puede establecer en `update_initial_cpb_removal_delay_offset[SchedSelIdx]`, en el que `initial_cpb_removal_delay[SchedSelIdx]`, `initial_cpb_removal_delay_offset[SchedSelIdx]`, `update_initial_cpb_removal_delay[SchedSelIdx]`, y `update_initial_cpb_removal_offset[SchedSelIdx]` se puede especificar en el mensaje SEI del período de almacenamiento en memoria intermedia asociado a la unidad de acceso CRA.

[0106] Además, la unidad de acceso que se asocia al mensaje SEI del período de almacenamiento en memoria intermedia que inicializa la CPB puede denominarse unidad de acceso 0. Todas las demás unidades de acceso se denominan unidad de acceso "n," y "n" se incrementa en 1 para la siguiente unidad de acceso en un orden de descodificación. En este ejemplo, el momento en el que el primer bit de la unidad de acceso n comienza a entrar en la CPB puede denominarse momento de llegada inicial $t_{ai}(m)$.

[0107] En un ejemplo, el momento de llegada inicial de las unidades de acceso puede obtenerse como se indica a continuación:

(1) Si la unidad de acceso es la unidad de acceso 0, $t_{ai}(0) = 0$;

(2) De otro modo (la unidad de acceso es la unidad de acceso n con $n > 0$), puede aplicarse lo siguiente:

(a) Si `cbr_flag[SchedSelIdx]` es igual a "1", el tiempo de llegada inicial para la unidad de acceso n, puede ser igual al tiempo de llegada final (que se obtiene a continuación) de la unidad de acceso n - 1, es decir,

$$t_{ai}(n) = t_{af}(n - 1) \quad \text{EC. 1}$$

5 (b) De otro modo, si `cbr_flag[SchedSelIdx]` es igual a "0" y la unidad de acceso `n` no es la primera unidad de acceso de un período de almacenamiento en memoria intermedia posterior, el tiempo de llegada inicial para la unidad de acceso `n` puede obtenerse mediante:

$$t_{ai}(n) = \text{Max}(t_{af}(n - 1), t_{ai,earliest}(n)) \quad \text{EC. 2}$$

10 donde $t_{ai,earliest}(n)$ se obtiene de la forma siguiente:

$$t_{ai,earliest}(n) = t_{r,n}(n) - (\text{InitialCpbRemovalDelay}[\text{SchedSelIdx}] + \text{InitialCpbRemovalDelayOffset}[\text{SchedSelIdx}] \div 90000) \quad \text{EC. 3}$$

15 siendo $t_{r,n}(n)$ el momento de eliminación nominal de la unidad de acceso `n` de la CPB, como se especifica en la subcláusula C.1.2 de HEVC WD4, en algunos ejemplos;

20 (c) De otro modo (si `cbr_flag[SchedSelIdx]` es igual a "0" y la unidad de acceso posterior `n` es la primera unidad de acceso de un período de almacenamiento en memoria intermedia posterior), el tiempo de llegada inicial para la unidad de acceso `n` puede obtenerse mediante:

$$t_{ai}(n) = t_{r,n}(n) - (\text{InitialCpbRemovalDelay}[\text{SchedSelIdx}] \div 90000) \quad \text{EC. 4}$$

25 con `InitialCpbRemovalDelay[SchedSelIdx]` especificado en el mensaje SEI del período de almacenamiento en memoria intermedia asociado a la unidad de acceso `n`, en algunos ejemplos.

[0108] En este ejemplo, el tiempo de llegada final para la unidad de acceso `n` puede obtenerse mediante:

$$t_{af}(n) = t_{ai}(n) + b(n) \div \text{BitRate}[\text{SchedSelIdx}] \quad \text{EC. 5}$$

30 donde $b(n)$ puede tener el tamaño en bits de la unidad de acceso `n`, contando los bits del flujo de bits de Tipo I para la conformidad de Tipo I, o los bits del flujo de bits de Tipo II para la conformidad de Tipo II.

[0109] En algunos ejemplos, los valores de `SchedSelIdx`, `BitRate[SchedSelIdx]`, y `CpbSize[SchedSelIdx]` pueden limitarse como se indica a continuación:

35 (1) si la unidad de acceso `n` y la unidad de acceso `n - 1` son parte de CVS diferentes, y el contenido de los SPS activos de los dos CVS es diferente, el HSS puede seleccionar un valor `SchedSelIdx1` de `SchedSelIdx` entre los valores de `SchedSelIdx` proporcionados para la CVS que contiene la unidad de acceso `n` que da como resultado una `BitRate[SchedSelIdx1]` o `CpbSize[SchedSelIdx1]` para el segundo de los dos CVS (que contiene la unidad de acceso `n - 1`) que difiere del valor de `BitRate[SchedSelIdx0]` o `CpbSize[SchedSelIdx0]` para el valor `SchedSelIdx0` de `SchedSelIdx` que estaba en uso para la CVS que contiene la unidad de acceso `n - 1`;

45 (2) de otro modo, el HSS puede continuar funcionando con los valores anteriores de `SchedSelIdx`, `BitRate[SchedSelIdx]` y `CpbSize[SchedSelIdx]`.

[0110] En otros ejemplos, cuando el HSS selecciona los valores de `BitRate[SchedSelIdx]` o `CpbSize[SchedSelIdx]` que difieren de los de la unidad de acceso anterior, puede aplicarse lo siguiente:

50 (1) la variable `BitRate[SchedSelIdx]` puede entrar en vigor en el momento $t_{ai}(n)$

(2) la variable `CpbSize[SchedSelIdx]` puede entrar en vigor como se indica a continuación:

55 (a) si el nuevo valor de `CpbSize[SchedSelIdx]` excede el tamaño de CPB anterior, puede entrar en vigor en el momento $t_{ai}(n)$,

(b) de lo contrario, el nuevo valor de `CpbSize[SchedSelIdx]` puede entrar en vigor en el momento $t_r(n)$.

[0111] Como otro ejemplo, ahora se describirá el momento de eliminación de la imagen codificada. Por ejemplo, para la unidad de acceso 0, el momento de eliminación nominal de la unidad de acceso desde la CPB puede especificarse mediante:

$$t_{r,n}(0) = \text{InitialCpbRemovalDelay}[\text{SchedSelIdx}] \div 90000 \quad \text{EC. 6}$$

[0112] Además, para la primera unidad de acceso de un período de almacenamiento en memoria intermedia que no inicializa el HRD, el momento de eliminación nominal de la unidad de acceso desde la CPB puede especificarse mediante:

$$t_{r,n}(n) = t_{r,n}(n_b) + t_c * cpb_removal_delay(n) \quad \text{EC. 7}$$

donde $t_{r,n}(n_b)$ puede ser el momento de eliminación nominal de la primera imagen del período anterior de almacenamiento en memoria intermedia, y $cpb_removal_delay(n)$ se puede especificar en el mensaje SEI de sincronización de imagen asociado con la unidad de acceso n.

[0113] Adicionalmente, cuando una unidad de acceso n es la primera unidad de acceso de un período de almacenamiento en memoria intermedia, n_b puede ajustarse igual a n en el momento de eliminación de la unidad de acceso m.

[0114] Igualmente, el momento de eliminación nominal $t_{r,n}(m)$ de una unidad de acceso n que no es la primera unidad de acceso de un período de almacenamiento en memoria intermedia puede obtenerse mediante:

$$t_{r,n}(n) = t_{r,n}(n_b) + t_c * cpb_removal_delay(n) \quad \text{EC. 8}$$

[0115] Además, el momento de eliminación de la unidad de acceso n puede especificarse como se indica a continuación:

(1) si $low_delay_hrd_flag$ es igual a "0" o $t_{r,n}(m) > t_{af}(m)$, el momento de eliminación de la unidad de acceso n puede especificarse mediante:

$$t_r(n) = t_{r,n}(n) \quad \text{EC. 9}$$

(2) de otro modo (si $low_delay_hrd_flag$ es igual a 1 y $t_{r,n}(n) < t_{af}(n)$), el momento de eliminación de la unidad de acceso n puede especificarse mediante:

$$t_r(n) = t_{r,n}(n) + t_c * Ceil((t_{af}(n) - t_{r,n}(n)) \div t_c) \quad \text{EC. 10}$$

[0116] En este ejemplo, el último caso indica que el tamaño de la unidad de acceso n, $b(n)$, es tan grande que se impide la eliminación en el momento de la eliminación nominal.

[0117] Como otro ejemplo, ahora se describirá la conformidad del flujo de bits. La divulgación de la subcláusula C.3 de H.264/AVC puede aplicarse a las técnicas descritas a continuación, con los siguientes cambios incluidos: una primera imagen codificada, en un orden de descodificación, de un flujo de bits conforme puede ser una imagen IDR o una imagen de CRA. Para un flujo de bits conforme que comienza con una imagen de CRA, un subconjunto de flujo de bits generado descartando unidades de acceso que contienen todas las imágenes principales asociadas con la imagen de CRA de inicio puede seguir siendo un flujo de bits conforme.

[0118] La siguiente descripción incluye ejemplos alternativos de las técnicas de esta divulgación descritas anteriormente. Por ejemplo, son posibles diversas implementaciones alternativas de ciertos aspectos de esta divulgación, y algunas se describen a continuación. Las siguientes implementaciones alternativas se describen con referencia a diferentes aspectos de las técnicas de la divulgación. Sin embargo, cualquier combinación de las implementaciones para los diferentes aspectos también puede formar implementaciones coherentes con las técnicas de esta divulgación.

[0119] Los siguientes ejemplos alternativos se relacionan con la modificación de un VPS.

[0120] Como un ejemplo, se proporciona una definición de aVPS de la siguiente manera: Un conjunto de imágenes de referencia asociadas con una imagen, que consta de todas las imágenes de referencia, que están en un RPS actual y pueden no descodificarse correctamente cuando se produce un acceso aleatorio desde una imagen de CRA más cercana, que precede a la imagen en un orden de descodificación.

[0121] El siguiente es un ejemplo alternativo de modificación de VPS. En este ejemplo, en el caso de que un VPS no esté vacío, se puede aplicar lo siguiente:

- (1) antes de descodificar una imagen actual, cada imagen en el VPS puede comprobarse;
 - (a) si la imagen está en el RPS, puede guardarse en el VPS;

(b) de otro modo, puede eliminarse del VPS;

(2) en el caso de que el VPS de una imagen actual y el RPS de la imagen actual se hayan superpuesto, la imagen actual se puede insertar en el VPS, si se trata de una imagen de referencia, después de descodificar la imagen actual.

[0122] En algunos ejemplos, si una imagen no es una imagen de CRA, y un VPS no está vacío, cada imagen en un RPS actual puede estar en el VPS o en el DPB.

[0123] El siguiente es otro ejemplo alternativo de modificación de VPS. En este ejemplo, en el caso de que un VPS no esté vacío, se puede aplicar lo siguiente:

(1) antes de descodificar una imagen, cada imagen en el VPS puede comprobarse;

(a) si la imagen está en el RPS y pertenece a uno de los subconjuntos RefPicSetStCurr0, RefPicSetStCurr1 y RefPicSetLtCurr, puede mantenerse en el VPS;

(b) de otro modo, puede eliminarse del VPS;

(2) en el caso de que el VPS de una imagen actual y el RPS de la imagen actual se hayan superpuesto, la imagen actual se puede insertar en el VPS, si se trata de una imagen de referencia, después de descodificar la imagen actual.

[0124] El siguiente es otro ejemplo alternativo de modificación de VPS. En este ejemplo, en el caso de que una imagen tenga un orden de visualización mayor que una imagen de CRA, y un VPS no esté vacío, se puede aplicar lo siguiente:

(1) en el caso de que al menos una imagen en un RPS pertenezca al VPS, el VPS puede mantenerse sin cambios;

(2) en el caso de que ninguna imagen en el RPS pertenezca al VPS, el VPS puede configurarse como vacío.

[0125] En este ejemplo, el VPS solo puede cambiar dos veces durante cada acceso aleatorio. La primera vez, puede llenarse con imágenes de las que pueden depender las imágenes principales. La segunda vez, puede configurarse como vacío.

[0126] En el ejemplo de la FIG. 4, si la imagen "8" se usa para acceso aleatorio, el VPS puede ser {8}. En este ejemplo, solo después de crear el RPS de la imagen "16", VPS puede configurarse como vacío. En este ejemplo, las imágenes con un orden de visualización inferior al de la imagen de CRA se pueden omitir para la descodificación y la salida, siempre que el VPS no esté vacío. Esto puede describirse como un enfoque diferente o alternativo para "Identificación de imágenes principales".

[0127] Ahora se describirá un ejemplo alternativo de creación de una imagen desaparecida. En este ejemplo, en el caso de que una imagen se detecte como una imagen desaparecida, puede crearse como una copia de una imagen en un DPB que tiene el orden de visualización más cercano (distancia POC) a la imagen desaparecida. Si dos imágenes tienen la misma distancia POC, se puede usar una con un POC más pequeño.

[0128] Ahora se describirá un ejemplo alternativo de descodificación de imagen principal. Como un ejemplo, una imagen principal puede descodificarse, si ninguna de las imágenes en RefPicSetStCurr0, RefPicSetStCurr1 o RefPicSetLtCurr pertenece al VPS. Como otro ejemplo, una imagen principal siempre puede descodificarse, especialmente cuando cada imagen desaparecida está disponible en el DPB, aunque con deriva.

[0129] Ahora se describirá un ejemplo alternativo de salida de imagen principal. Como ejemplo, en el caso de que una imagen principal se descodifique correctamente, output_flag se puede establecer en "1". De lo contrario, el indicador de salida se puede establecer en "0". Como otro ejemplo, solo las imágenes principales continuas inmediatamente antes de la imagen de CRA en un orden de salida, si están todas descodificadas correctamente, pueden tener valores de indicador de salida establecidos en "1", y otras imágenes principales pueden tener valores de indicador de salida establecidos en "0".

[0130] Ahora se describirá un ejemplo alternativo del desplazamiento del momento de eliminación, y en particular, la sintaxis del mensaje SEI del período de almacenamiento en memoria intermedia.

TABLA III

buffering_period(payloadSize) {	C	Descriptor
seq_parameter_set_id	5	ue(v)
cra_para_present_flag	5	u(l)
if(NalHrdBpPresentFlag) {		
for(SchedSelIdx = 0; SchedSelIdx <= cpb_cnt_minus1; SchedSelIdx++) {		
initial_cpb_removal_delay[SchedSelIdx]	5	u(v)
initial_cpb_removal_delay_offset[SchedSelIdx]	5	u(v)
if (cra_para_present_flag)		
random_access_removal_delay_offset[SchedSelIdx]	5	u(v)
}		
}		
if(VclHrdBpPresentFlag) {		
for(SchedSelIdx = 0; SchedSelIdx <= cpb_cnt_minus1; SchedSelIdx++) {		
initial_cpb_removal_delay[SchedSelIdx]	5	u(v)
initial_cpb_removal_delay_offset[SchedSelIdx]	5	u(v)
if(cra_para_present_flag)		
random_access_removal_delay_offset[SchedSelIdx]	5	u(v)
buffering_period(payloadSize) {	C	Descriptor
}		
}		
}		

5

[0131] De forma alternativa, en otros ejemplos, el desplazamiento puede señalarse en un mensaje SEI diferente que solo está asociado con una imagen de CRA (o acceso aleatorio no IDR), para la cual se proporciona la sintaxis a continuación.

10

[0132] Se describirá ahora un ejemplo de la sintaxis del mensaje SEI de compensación de retardo de eliminación de CPB.

Tabla IV

cpb_removeal_delay_offset(payloadSize) {	C	Descriptor
seq_parameter_set_id	5	ue(v)
if(NalHrdBpPresentFlag)		
for(SchedSelIdx = 0; SchedSelIdx <= cpb_cnt_minus1; SchedSelIdx++)		
random_access_removal_delay_offset[SchedSelIdx]	5	u(v)
if(VclHrdBpPresentFlag)		
for(SchedSelIdx = 0; SchedSelIdx <= cpb_cnt_minus1; SchedSelIdx++) {		
random_access_removal_delay_offset[SchedSelIdx]	5	u(v)
}		

15

[0133] Además, la semántica del mensaje SEI del período de almacenamiento en memoria intermedia se describe a continuación.

5 **[0134]** Como un ejemplo, `cra_para_present_flag` igual a "1" puede indicar la presencia del elemento sintáctico `random_access_removal_delay_offset[SchedSelIdx]`. Este indicador igual a "0" puede indicar la ausencia del elemento sintáctico `random_access_removal_delay_offset[SchedSelIdx]`.

10 **[0135]** Como otro ejemplo, `random_access_removal_delay_offset[SchedSelIdx]` puede especificar una compensación de momento de eliminación de CPB para la `SchedSelIdx`-ésima CPB. Por ejemplo, puede ser en unidades de un reloj de 90 kHz. Además, `random_access_removal_delay_offset[SchedSelIdx]` no puede exceder `initial_cpb_removal_delay[SchedSelIdx] + initial_cpb_removal_delay_offset[SchedSelIdx]`. En algunos ejemplos, cuando no está presente, se puede inferir que el valor es igual a "0".

15 **[0136]** Se describirá ahora un ejemplo de la semántica del mensaje SEI de compensación de retardo de eliminación de CPB. En algunos ejemplos, dicho mensaje SEI solo puede estar presente para una imagen de CRA, y solo puede tener efecto cuando el CRA se usa para acceso aleatorio y sus correspondientes imágenes principales no están presentes en el flujo de bits. Como un ejemplo, `random_access_removal_delay_offset[SchedSelIdx]` puede especificar una compensación de momento de eliminación de CPB para la `SchedSelIdx`-ésima CPB. Por ejemplo, puede ser en unidades de un reloj de 90 kHz. Además, `random_access_removal_delay_offset[SchedSelIdx]` no puede exceder `initial_cpb_removal_delay[SchedSelIdx] + initial_cpb_removal_delay_offset[SchedSelIdx]`. En algunos ejemplos, cuando no está presente, se puede inferir que el valor es igual a "0".

25 **[0137]** Se describirá ahora un ejemplo de funcionamiento de una CPB. Las especificaciones en esta parte de la divulgación se pueden aplicar de forma independiente a cada conjunto de parámetros CPB que está presente y a la conformidad tanto de Tipo I como de Tipo II.

30 **[0138]** Como un ejemplo, ahora se describirá el tiempo de llegada del flujo de bits. Por ejemplo, el HRD puede inicializarse en uno cualquiera de los mensajes SEI del período de almacenamiento en memoria intermedia. Antes de la inicialización, la CPB puede estar vacía. En este ejemplo, después de la inicialización, el HRD puede no inicializarse de nuevo mediante los posteriores mensajes SEI del período de almacenamiento en memoria intermedia. También en este ejemplo, la unidad de acceso que se asocia al mensaje SEI del período de almacenamiento en memoria intermedia que inicializa la CPB puede denominarse la unidad de acceso 0. Todas las demás unidades de acceso se denominan unidad de acceso `n`, donde `n` se incrementa en 1 para la siguiente unidad de acceso en un orden de descodificación.

35 **[0139]** En este ejemplo, si la primera unidad de acceso es una unidad de acceso CRA, y las imágenes principales no están presentes y el `cra_para_present_flag` es igual a "1", `useUpdatePara` puede establecerse en "1". De lo contrario, `useUpdatePara` se puede establecer en "0". Además, si `useUpdatePara` es igual a "1", `DelayOffset[SchedSelIdx]` se puede establecer en `random_access_removal_delay_offset[SchedSelIdx]`. De lo contrario, `DelayOffset[SchedSelIdx]` se puede establecer en "0". Adicionalmente, el momento en el que el primer bit de la unidad de acceso `n` comienza a entrar en la CPB puede denominarse el tiempo de llegada inicial $t_{ai}(n)$. En algunos ejemplos, el momento de llegada inicial de las unidades de acceso puede obtenerse como se indica a continuación:

(1) Si la unidad de acceso es la unidad de acceso 0, $t_{ai}(0) = 0$;

(2) De otro modo (si la unidad de acceso es la unidad de acceso `n` con `n > 0`), puede aplicarse lo siguiente:

50 (a) Si `cbr_flag[SchedSelIdx]` es igual a "1", el tiempo de llegada inicial para la unidad de acceso `n`, puede ser igual al tiempo de llegada final (que se obtiene a continuación) de la unidad de acceso `n - 1`, es decir,

$$t_{ai}(n) = t_{af}(n - 1) \quad \text{EC. 11}$$

55 (b) De otro modo, si `cbr_flag[SchedSelIdx]` es igual a "0" y la unidad de acceso `n` no es la primera unidad de acceso de un período de almacenamiento en memoria intermedia posterior, el tiempo de llegada inicial para la unidad de acceso `n` puede obtenerse mediante:

$$t_{ai}(n) = \text{Max}(t_{af}(n - 1), t_{ai,earliest}(n)) \quad \text{EC. 12}$$

60 donde $t_{ai,earliest}(n)$ puede obtenerse como se indica a continuación:

$$t_{ai,earliest}(n) = t_{r,n}(n) - (\text{initial_cpb_removal_delay}[\text{SchedSelIdx}] + \text{initial_cpb_removal_delay_offset}[\text{SchedSelIdx}] \div 90000) \quad \text{EC. 13}$$

siendo $t_{r,n}(n)$ el momento de eliminación nominal de la unidad de acceso n de la CPB como se especifica en la sub-cláusula C.1.2 de HEVC WD4, e $\text{initial_cpb_removal_delay}[\text{SchedSelIdx}]$ e $\text{initial_cpb_removal_delay_offset}[\text{SchedSelIdx}]$ especificándose en el mensaje SEI del período de almacenamiento en memoria intermedia anterior.

(c) De otro modo (si $\text{cbr_flag}[\text{SchedSelIdx}]$ es igual a "0" y la unidad de acceso posterior n es la primera unidad de acceso de un período de almacenamiento en memoria intermedia posterior), el tiempo de llegada inicial para la unidad de acceso n puede obtenerse mediante:

$$t_{ai}(n) = t_{r,n}(n) - (\text{initial_cpb_removal_delay}[\text{SchedSelIdx}] \div 90000) \quad \text{EC. 14}$$

con $\text{initial_cpb_removal_delay}[\text{SchedSelIdx}]$ que se especifica en el mensaje SEI del período de almacenamiento en memoria intermedia asociado a la unidad de acceso n . En este ejemplo, el tiempo de llegada final para la unidad de acceso n puede obtenerse mediante:

$$t_{af}(n) = t_{ai}(n) + b(n) \div \text{BitRate}[\text{SchedSelIdx}] \quad \text{EC. 15}$$

donde $b(n)$ puede tener el tamaño en bits de la unidad de acceso n , contando los bits del flujo de bits de Tipo I para la conformidad de Tipo I, o los bits del flujo de bits de Tipo II para la conformidad de Tipo II.

[0140] Además, los valores de SchedSelIdx , $\text{BitRate}[\text{SchedSelIdx}]$, y $\text{CpbSize}[\text{SchedSelIdx}]$ pueden limitarse como se indica a continuación:

(1) Si la unidad de acceso n y la unidad de acceso " $n - 1$ " son parte de CVS diferentes, y el contenido de los SPS activos de los dos CVS es diferente, el HSS puede seleccionar un valor SchedSelIdx1 de SchedSelIdx entre los valores de SchedSelIdx provistos para la CVS que contiene la unidad de acceso n que da como resultado un $\text{BitRate}[\text{SchedSelIdx1}]$ o $\text{CpbSize}[\text{SchedSelIdx1}]$ para el segundo de los dos CVS (que contiene la unidad de acceso $n - 1$) que difiere del valor de $\text{BitRate}[\text{SchedSelIdx0}]$ o " $\text{CpbSize}[\text{SchedSelIdx0}]$ " para el valor SchedSelIdx0 de SchedSelIdx que estaba en uso para la CVS que contiene la unidad de acceso $n - 1$.

(2) De otro modo, el HSS puede continuar funcionando con los valores anteriores de SchedSelIdx , $\text{BitRate}[\text{SchedSelIdx}]$ y $\text{CpbSize}[\text{SchedSelIdx}]$.

[0141] Adicionalmente, cuando el HSS selecciona los valores de $\text{BitRate}[\text{SchedSelIdx}]$ o $\text{CpbSize}[\text{SchedSelIdx}]$ que difieren de los de la unidad de acceso anterior, puede aplicarse lo siguiente:

(1) La variable $\text{BitRate}[\text{SchedSelIdx}]$ puede entrar en vigor en el momento $t_{ai}(n)$; y

(2) La variable $\text{CpbSize}[\text{SchedSelIdx}]$ puede entrar en vigor como se indica a continuación:

(a) Si el nuevo valor de $\text{CpbSize}[\text{SchedSelIdx}]$ excede el tamaño de CPB anterior, puede entrar en vigor en el momento $t_{ai}(n)$,

(b) De lo contrario, el nuevo valor de $\text{CpbSize}[\text{SchedSelIdx}]$ puede entrar en vigor en el momento $t_r(n)$.

[0142] Como otro ejemplo, ahora se describirá el momento de eliminación de la imagen codificada. En algunos ejemplos, se puede suponer que el momento de eliminación de CPB nominal y el momento de eliminación de CPB de una imagen codificada se calculan inmediatamente después de que la imagen codificada anterior se elimina de la CPB o, para la unidad de acceso 0, cuando se inicializa el HRD.

[0143] Por ejemplo, para la unidad de acceso 0, el momento de eliminación nominal de la unidad de acceso desde la CPB puede especificarse mediante:

$$t_{r,n}(0) = \text{initial_cpb_removal_delay}[\text{SchedSelIdx}] \div 90000 \quad \text{EC. 16}$$

[0144] En el momento de la eliminación de la unidad de acceso 0, la variable n_b se puede configurar igual a "0". Inmediatamente después de la eliminación de la unidad de acceso 0 de la CPB, $t_{r,n}(0)$ se puede configurar igual a $t_{r,n}(0) - (\text{DelayOffset}[\text{SchedSelIdx}] \div 90000)$.

[0145] En este ejemplo, el momento efectivo de eliminación de CPB de la unidad de acceso 0 no se puede desplazar, pero para todas las imágenes después de la unidad de acceso 0 en un orden de decodificación, el momento efectivo de eliminación de CPB puede cambiarse antes en $(\text{DelayOffset}[\text{SchedSelIdx}] \div 90000)$.

5 **[0146]** Además, para la primera unidad de acceso de un período de almacenamiento en memoria intermedia que no inicializa el HRD, el momento de eliminación nominal de la unidad de acceso desde la CPB puede especificarse mediante:

$$t_{r,n}(n) = t_{r,n}(n_b) + t_c * \text{cpb_removal_delay}(n) \quad \text{EC. 17}$$

10

donde $t_{r,n}(n_b)$ puede ser el momento de eliminación nominal de la primera imagen del período anterior de almacenamiento en memoria intermedia, y $\text{cpb_removal_delay}(n)$ se puede especificar en el mensaje SEI de sincronización de imagen asociado con la unidad de acceso n.

15 **[0147]** Además, cuando una unidad de acceso n es la primera unidad de acceso de un período de almacenamiento en memoria intermedia que no inicializa el HRD, n_b puede establecerse igual a "n" en el momento de eliminación de la unidad de acceso n. Además, el momento de eliminación nominal $t_{r,n}(n)$ de una unidad de acceso n que no es la primera unidad de acceso de un período de almacenamiento temporal puede venir dado por:

20

$$t_{r,n}(n) = t_{r,n}(n_b) + t_c * \text{cpb_removal_delay}(n) \quad \text{EC. 18}$$

[0148] Por ejemplo, el momento de eliminación de la unidad de acceso n puede especificarse como se indica a continuación:

25

(1) Si $\text{low_delay_hrd_flag}$ es igual a "0", o $t_{r,n}(n) \geq t_{af}(n)$, el momento de eliminación de la unidad de acceso n puede especificarse mediante:

$$t_r(n) = t_{r,n}(n) \quad \text{EC. 19}$$

30

(2) De otro modo (si $\text{low_delay_hrd_flag}$ es igual a "1" y $t_{r,n}(n) < t_{af}(n)$), el momento de eliminación de la unidad de acceso n puede especificarse mediante:

$$t_r(n) = t_{r,n}(n) + t_c * \text{Ceil}((t_{af}(n) - t_{r,n}(n)) \div t_c) \quad \text{EC. 20}$$

35

[0149] En este ejemplo, el último caso indica que el tamaño de la unidad de acceso n, $b(n)$, es tan grande que se impide la eliminación en el momento de eliminación nominal.

40 **[0150]** Otro ejemplo de las técnicas de esta divulgación especifica procesos de decodificación para imágenes principales que tienen imágenes de referencia faltantes. En este ejemplo, solo un flujo de bits que comienza con una imagen de CRA para el que están presentes las imágenes principales en el flujo de bits se especifica como un flujo de bits conforme. En particular, este ejemplo corresponde a algunas de las técnicas descritas en HEVC WD5, e incluye los siguientes cambios a estas técnicas de HEVC WD5, como se describe en detalle a continuación.

45 **[0151]** En este ejemplo, una primera imagen codificada en un flujo de bits puede ser una imagen IDR o una imagen de CRA. Como se explicó anteriormente, el término "imagen principal", como se usa en esta divulgación, se puede definir de la siguiente manera: Una imagen codificada asociada con una imagen de CRA que sigue a la imagen de CRA en un orden de decodificación, y precede a la imagen de CRA en un orden de salida. Por ejemplo, si la primera imagen codificada en el flujo de bits es una imagen de CRA, y la imagen actual, o actualmente codificada, es una imagen principal de la primera imagen codificada en el flujo de bits, output_flag de la imagen codificada actualmente puede configurarse para ser igual a falso, o "0" (por ejemplo, independientemente del valor del output_flag en la cabecera de la unidad NAL de las unidades NAL de la capa de codificación de vídeo (VCL) de la imagen codificada). En este ejemplo, se puede invocar el proceso de decodificación para generar imágenes de referencia faltantes (como se muestra a continuación) (por ejemplo, tal vez solo sea necesario invocar este proceso para un fragmento de una imagen).

50 **[0152]** Además, puede haber una o más imágenes de referencia incluidas en el RPS, pero que no están presentes en el DPB. Las entradas en RefPicSetStFoll o RefPicSetLtFoll iguales a "sin imagen de referencia" pueden ignorarse si la primera imagen codificada en el flujo de bits es una imagen IDR, o si la primera imagen codificada en el flujo de bits es una imagen de CRA y la imagen codificada actualmente no es una imagen principal de la primera imagen codificada en el flujo de bits. Por ejemplo, se puede inferir una pérdida de imagen involuntaria para cada entrada en RefPicSetStCurr0 , RefPicSetStCurr1 y RefPicSetLtCurr igual a "sin imagen de referencia".

60

[0153] Además, si la primera imagen codificada en el flujo de bits es una imagen IDR, o si la primera imagen codificada en el flujo de bits es una imagen de CRA y la imagen codificada actualmente no es una imagen principal de la primera imagen codificada en el flujo de bits, tal vez no haya entrada en RefPicSetStCurr0, RefPicSetStCurr1, o RefPicSetLtCurr igual a "sin imagen de referencia".

[0154] El proceso de decodificación para generar imágenes de referencia faltantes se puede especificar de la siguiente manera. Este proceso puede invocarse una vez por cada imagen codificada, después de la invocación del proceso de decodificación para RPS (como se especifica en la subcláusula 8.2.2 en HEVC WD5, en el documento "JCTVC-G1103", por ejemplo, en la versión "d9" del documento). Cuando la primera imagen codificada en el flujo de bits es una imagen de CRA, y la imagen codificada actualmente es una imagen principal de la primera imagen codificada en el flujo de bits, puede aplicarse lo siguiente:

1) Para cada RefPicSetStCurr0[i], con "i" en el rango de "0" a NumPocStCurr0 - 1, inclusive, que es igual a "sin imagen de referencia", se puede generar una imagen de referencia mediante la invocación del proceso de decodificación para generar una imagen de referencia faltante, como se especifica a continuación. Además, puede aplicarse lo siguiente:

- A) El valor de PicOrderCntVal para la imagen de referencia generada se puede establecer en PocStCurr0[i].
- B) El valor de output_flag para la imagen de referencia generada se puede establecer en "0".
- C) La imagen de referencia generada se puede marcar como "utilizada para referencia a corto plazo".
- D) RefPicSetStCurr0[i] se puede configurar para que sea la imagen de referencia generada.

2) Para cada RefPicSetStCurr1[i], con "i" en el rango de "0" a NumPocStCurr1 - 1, inclusive, que es igual a "sin imagen de referencia", se puede generar una imagen de referencia mediante la invocación del proceso de decodificación para generar una imagen de referencia faltante, como se especifica a continuación. Además, puede aplicarse lo siguiente:

- A) El valor de PicOrderCntVal para la imagen de referencia generada se puede establecer en PocStCurr1[i].
- B) El valor del indicador de salida para la imagen de referencia generada se puede establecer en "0".
- C) La imagen de referencia generada se puede marcar como "utilizada para referencia a corto plazo".
- D) RefPicSetStCurr1[i] se puede configurar para que sea la imagen de referencia generada.

3) Para cada RefPicSetLtCurr[i], con "i" en el rango de "0" a NumPocLtCurr - 1, inclusive, que es igual a "sin imagen de referencia", se puede generar una imagen de referencia mediante la invocación del proceso de decodificación para generar una imagen de referencia faltante, como se especifica a continuación. Además, puede aplicarse lo siguiente:

- A) El valor de pic_order_cnt_lsb para la imagen de referencia generada se puede establecer en PocLtCurr[i].
- B) El valor de output_flag para la imagen de referencia generada se puede establecer en "0".
- C) La imagen de referencia generada puede marcarse como "utilizada para referencia a largo plazo".
- D) RefPicSetLtCurr[i] se puede configurar para que sea la imagen de referencia generada.

[0155] En algunos ejemplos, el proceso de decodificación para generar una imagen de referencia faltante se puede especificar de la siguiente manera:

- 1) El valor de cada elemento en el conjunto de muestras S_L se puede establecer en $1 \ll (\text{BitDepth}_Y - 1)$.
- 2) El valor de cada elemento en las matrices de muestra S_{Cb} y S_{Cr} puede establecerse en $1 \ll (\text{BitDepth}_C - 1)$.
- 3) El modo de predicción PredMode para cada CU mínima puede establecerse en MODE_INTRA.

[0156] De forma alternativa, cada LCU se puede dividir en una CU mínima, y la CU mínima se puede establecer en MODE_INTRA.

[0157] De forma alternativa, cada LCU puede configurarse para ser MODE_INTRA.

[0158] En consecuencia, en algunos ejemplos coherentes con las técnicas de esta divulgación, el codificador de vídeo 20 del dispositivo de origen 12 puede configurarse para codificar una o más imágenes de datos de vídeo. En estos ejemplos, el descodificador de vídeo 30 del dispositivo de destino 14 puede configurarse para recibir la una o más imágenes codificadas del codificador de vídeo 20, por ejemplo, como parte de un flujo de bits codificado generado por el codificador de vídeo 20 y recibido por el descodificador de vídeo 30 y descodificar la una o más imágenes

[0159] Como un ejemplo, el descodificador de vídeo 30 puede configurarse para recibir un flujo de bits que incluye una o más imágenes de una CVS. El descodificador de vídeo 30 puede configurarse adicionalmente para descodificar una primera imagen de la una o más imágenes de acuerdo con un orden de descodificación asociado con la CVS. En este ejemplo, la primera imagen puede ser una imagen RAP que no es una imagen IDR. El descodificador de vídeo 30 también puede configurarse para descodificar al menos una de las una o más imágenes, distintas de la primera imagen, siguiendo a la primera imagen de acuerdo con el orden de descodificación, basándose en la primera imagen descodificada.

[0160] Como otro ejemplo, el codificador de vídeo 20 puede configurarse para generar un flujo de bits que incluye una o más imágenes de una CVS. En este ejemplo, una primera imagen de la una o más imágenes de acuerdo con un orden de descodificación asociado con la CVS puede ser una imagen RAP que no es una imagen IDR. También en este ejemplo, para generar el flujo de bits, el codificador de vídeo 20 puede configurarse para evitar incluir al menos una de las una o más imágenes, además de la primera imagen, que corresponde a una imagen principal asociada con la primera imagen, en el flujo de bits. Por ejemplo, la imagen principal puede ser una imagen que sigue a la primera imagen de acuerdo con el orden de descodificación y precede a la primera imagen de acuerdo con un orden de visualización asociado con la CVS. En este ejemplo, la primera imagen puede ser descodificable, es decir, puede descodificarse, por ejemplo, mediante el descodificador de vídeo 30. También en este ejemplo, al menos una de las una o más imágenes, además de la primera imagen, que sigue a la primera imagen de acuerdo con el orden de descodificación, también puede ser descodificable, basándose en la primera imagen descodificada. Por ejemplo, la al menos una de las una o más imágenes, distintas de la primera imagen, que siguen a la primera imagen de acuerdo con el orden de descodificación pueden ser descodificables usando una versión descodificada de la primera imagen como una imagen de referencia.

[0161] De esta manera, el descodificador de vídeo 30 puede descodificar un flujo de bits, por ejemplo, generado por el codificador de vídeo 20, que incluye una o más imágenes de datos de vídeo y comienza con una imagen RAP no IDR, de una manera predecible y definida, según lo especificado por las técnicas de esta divulgación. Como resultado, puede haber una mejora relativa en la experiencia del usuario cuando se realiza un acceso aleatorio de un flujo de bits que incluye una o más imágenes de datos de vídeo, cuando se usan las técnicas divulgadas. En particular, el descodificador de vídeo 30 puede descodificar el flujo de bits con una granularidad relativamente mayor. En otras palabras, el descodificador de vídeo 30 puede acceder aleatoriamente al flujo de bits en relativamente más puntos, o imágenes (es decir, imágenes no IDR) del flujo de bits, en comparación con otras técnicas (por ejemplo, técnicas que permiten el acceso aleatorio de un flujo de bits solamente a partir de imágenes IDR. Adicionalmente, puede haber una mejora relativa en la calidad visual de una o más imágenes de una CVS incluida en el flujo de bits, y/o de la CVS como un todo, cuando se usan las técnicas de esta divulgación.

[0162] Cada uno del codificador de vídeo 20 y el descodificador de vídeo 30 pueden implementarse como cualquiera entre una amplia variedad de circuitos codificadores o descodificadores adecuados, según corresponda, tales como uno o más microprocesadores, DSP, ASIC, FPGA, circuitos de lógica discreta, software, hardware, firmware o cualquier combinación de los mismos. Cada uno del codificador de vídeo 20 y el descodificador de vídeo 30 pueden estar incluidos en uno o más codificadores o descodificadores, cada uno de los cuales puede estar integrado como parte de un codificador/descodificador (CÓDEC) de vídeo combinado. Un aparato que incluye un codificador de vídeo 20 y/o un descodificador de vídeo 30 puede comprender un circuito integrado (CI), un microprocesador y/o un dispositivo de comunicación inalámbrica, tal como un teléfono celular.

[0163] La FIG. 2 es un diagrama de bloques que ilustra un ejemplo de un codificador de vídeo que puede implementar las técnicas para acceso aleatorio con gestión de DPB avanzada, coherentes con las técnicas de esta divulgación. El codificador de vídeo 20 puede realizar la intra-codificación y la inter-codificación de bloques de vídeo dentro de fragmentos de vídeo. La intra-codificación se basa en la predicción espacial para reducir o eliminar la redundancia espacial en el vídeo dentro de una trama o imagen de vídeo dada. La inter-codificación se basa en la predicción temporal para reducir o eliminar la redundancia temporal en el vídeo dentro de tramas o imágenes adyacentes de una secuencia de vídeo. El modo intra (modo I) puede referirse a cualquiera de varios modos de compresión espacial. Los modos inter, tales como la predicción unidireccional (modo P) o la predicción bidireccional (modo B), pueden referirse a cualquiera de varios modos de compresión temporal.

[0164] En el ejemplo de la FIG. 2, el codificador de vídeo 20 incluye una unidad de selección de modo 40, una unidad de estimación de movimiento 42, una unidad de compensación 44, una unidad de procesamiento de predicción intra 46, una memoria de imágenes de referencia 66, un sumador 50, una unidad de procesamiento de transformación 52, una unidad de cuantificación 54 y una unidad de codificación por entropía 56. Para la reconstrucción de bloques de vídeo, el codificador de vídeo 20 incluye también una unidad de cuantificación inversa

58, una unidad de procesamiento de transformación inversa 60 y un sumador 62. También se incluye un filtro de desbloqueo 64 para filtrar límites de bloque, para eliminar distorsiones de formación de bloques del vídeo reconstruido.

5 **[0165]** Como se muestra en la FIG. 2, el codificador de vídeo 20 recibe un bloque de vídeo actual dentro de un fragmento de vídeo que se va a codificar. El fragmento puede dividirse en múltiples bloques de vídeo. La unidad de selección de modo 40 puede seleccionar uno de los modos de codificación, intra o inter, para el bloque de vídeo actual, basándose en los resultados de error. Si se seleccionan los modos inter o intra, la unidad de selección de modo 40 proporciona el bloque intra-codificado o inter-codificado resultante al sumador 50 para generar datos de bloque residual y al sumador 62 para reconstruir el bloque codificado para usar como una imagen de referencia. 10 La unidad de procesamiento de predicción intra 46 lleva a cabo la codificación intra-predictiva del bloque de vídeo actual con respecto a uno o más bloques contiguos en la misma trama o fragmento que el bloque que va a codificarse, para proporcionar compresión espacial. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 realizan una codificación interpredictiva del bloque de vídeo actual con respecto a uno o más bloques predictivos de una o más imágenes de referencia para proporcionar una compresión temporal. 15

[0166] En el caso de codificación inter, la unidad de estimación de movimiento 42 puede estar configurada para determinar el modo de predicción inter para un fragmento de vídeo de acuerdo con un patrón por defecto para una secuencia de vídeo. El patrón por defecto puede designar fragmentos de vídeo de la secuencia como fragmentos P, fragmentos B o fragmentos GPB. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden estar altamente integradas, pero se ilustran por separado con fines conceptuales. La estimación de movimiento, realizada por la unidad de estimación de movimiento 42, es el proceso de generación de vectores de movimiento, que estiman el movimiento de los bloques de vídeo. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de una PU de un bloque de vídeo de una trama o imagen de vídeo actual con respecto a un bloque predictivo de una imagen de referencia. 20 25

[0167] Un bloque predictivo es un bloque del que se descubre que se corresponde estrechamente con la PU del bloque de vídeo que se va a codificar en términos de diferencia de píxeles, que puede determinarse mediante la suma de una diferencia absoluta (SAD), suma de diferencia al cuadrado (SSD) u otras métricas de diferencia. En algunos ejemplos, el codificador de vídeo 20 puede calcular valores para posiciones fraccionarias de píxeles de imágenes de referencia almacenadas en la memoria de imágenes de referencia 66. Por ejemplo, el codificador de vídeo 20 puede calcular valores de posiciones de un cuarto de píxel, posiciones de un octavo de píxel u otras posiciones de fracciones de píxel de la imagen de referencia. Por lo tanto, la unidad de estimación de movimiento 42 puede realizar una búsqueda de movimiento en relación con las posiciones de píxeles completas y las posiciones de píxeles fraccionarias, y generar un vector de movimiento con una precisión de píxel fraccionaria. 30 35

[0168] La unidad de estimación de movimiento 42 calcula un vector de movimiento para una PU de un bloque de vídeo en un fragmento inter-codificado, comparando la posición de la PU con la posición de un bloque predictivo de una imagen de referencia. La imagen de referencia puede seleccionarse de una primera lista de imágenes de referencia (Lista 0) o de una segunda lista de imágenes de referencia (Lista 1), cada una de las cuales identifica una o más imágenes de referencia almacenadas en la memoria de imágenes de referencia 66. La unidad de estimación de movimiento 42 envía el vector de movimiento calculado a la unidad de codificación por entropía 56 y a la unidad de compensación de movimiento 44. 40

[0169] La compensación de movimiento, llevada a cabo mediante la unidad de compensación de movimiento 44, puede implicar capturar o generar el bloque predictivo basándose en el vector de movimiento determinado por la estimación de movimiento. Tras recibir el vector de movimiento para la PU del bloque de vídeo actual, la unidad de compensación de movimiento 44 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 píxeles del bloque predictivo a los valores de píxeles del bloque de vídeo actual que se está codificando, generando valores de diferencia de píxel. Los valores de diferencia de píxel forman datos residuales para el bloque, y pueden incluir componentes de diferencia de luma y croma. El sumador 50 representa el componente o los componentes que realizan esta operación de resta. La unidad de compensación de movimiento 44 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. 45 50 55

[0170] Después de que la unidad de compensación de movimiento 44 genere el bloque predictivo para el bloque de vídeo actual, el codificador de vídeo 20 forma un bloque de vídeo residual al restar el bloque predictivo del bloque de vídeo actual. Los datos de vídeo residuales del bloque residual pueden incluirse en una o más TU y aplicarse a la unidad de procesamiento de transformación 52. La unidad de procesamiento de transformación 52 transforma los datos de vídeo residuales en coeficientes de transformación residuales mediante una transformación, tal como una transformación de coseno discreta (DCT) o una transformación similar desde un punto de vista conceptual. La unidad de procesamiento de transformación 52 puede convertir los datos de vídeo residuales de un dominio de píxeles a un dominio de las transformaciones, tal como un dominio de frecuencia. 60 65

[0171] La unidad de procesamiento de transformación 52 puede enviar los coeficientes de transformación resultantes a la unidad de cuantificación 54. La unidad de cuantificación 54 cuantifica los coeficientes de transformación para reducir adicionalmente la velocidad de transmisión de bits. El proceso de cuantificación puede reducir la profundidad de bits asociada con algunos o la totalidad de los coeficientes. El grado de cuantificación puede modificarse ajustando un parámetro de cuantificación. En algunos ejemplos, la unidad de cuantificación 54 puede realizar, a continuación, un escaneado de la matriz que incluye los coeficientes de transformación cuantificados. De forma alternativa, la unidad de codificación por entropía 56 puede llevar a cabo el escaneado.

[0172] Tras la cuantificación, la unidad de codificación por entropía 56 codifica por entropía los coeficientes de transformación cuantificados. Por ejemplo, la unidad de codificación por entropía 56 puede realizar CAVLC, CABAC u otra técnica de codificación por entropía. Tras la codificación por entropía realizada por la unidad de codificación por entropía 56, el flujo de bits codificado puede transmitirse al descodificador de vídeo 30, o archivar para su posterior transmisión o recuperación por el descodificador de vídeo 30. La unidad de codificación por entropía 56 también puede realizar la codificación por entropía de los vectores de movimiento y los otros elementos sintácticos para el fragmento de vídeo actual que se está codificando.

[0173] La unidad de cuantificación inversa 58 y la unidad de procesamiento de transformación inversa 60 aplican una cuantificación inversa y una transformación 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 44 puede calcular un bloque de referencia sumando 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 44 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 predictivo con compensación de movimiento generado por la unidad de compensación de movimiento 44 para generar un bloque de referencia para su almacenamiento en la memoria de imágenes de referencia 66. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden usar el bloque de referencia como un bloque de referencia para realizar la predicción inter de un bloque en una imagen o trama de vídeo subsiguiente.

[0174] Como un ejemplo, el codificador de vídeo 20 puede configurarse para codificar una o más imágenes de datos de vídeo durante un proceso de codificación de vídeo. Por ejemplo, el codificador de vídeo 20 puede configurarse para generar un flujo de bits que incluye una o más imágenes de una CVS, en el que una primera imagen de la una o más imágenes de acuerdo con un orden de descodificación asociado con la CVS es una imagen RAP que no es una imagen IDR. En este ejemplo, para generar el flujo de bits, el codificador de vídeo 20 puede configurarse para evitar incluir al menos una de las una o más imágenes, distintas de la primera imagen, que corresponde a una imagen principal asociada con la primera imagen en el flujo de bits. Por ejemplo, la imagen principal puede ser una imagen que sigue a la primera imagen de acuerdo con el orden de descodificación y precede a la primera imagen de acuerdo con un orden de visualización asociado con la CVS. También en este ejemplo, como resultado del codificador de vídeo 20 que genera el flujo de bits de la manera descrita anteriormente, la primera imagen puede descodificarse con éxito (por ejemplo, mediante el descodificador de vídeo 30), es decir, puede descodificarse. Además, al menos una de las una o más imágenes, además de la primera imagen, que sigue a la primera imagen de acuerdo con el orden de descodificación, también puede descodificarse con éxito (por ejemplo, mediante un descodificador de vídeo 30), o ser descodificable, basándose en la primera imagen (por ejemplo, usando la primera imagen como imagen de referencia, después de que la primera imagen haya sido descodificada como se describió anteriormente).

[0175] Por consiguiente, como se explicó anteriormente, las técnicas de esta divulgación pueden permitir que el codificador de vídeo 20 genere un flujo de bits que puede descodificarse mediante un descodificador de vídeo, por ejemplo, descodificador de vídeo 30, de una manera predecible y definida, según lo especificado por las técnicas de esta divulgación. En particular, el flujo de bits puede incluir una o más imágenes de una CVS de datos de vídeo. El descodificador de vídeo puede recibir el flujo de bits de manera que el flujo de bits comience con una imagen RAP no IDR. Usando las técnicas de esta divulgación, el descodificador de vídeo puede descodificar con éxito el flujo de bits. Como tal, puede haber una mejora relativa en la experiencia del usuario cuando se realiza un acceso aleatorio del flujo de bits, cuando se usan las técnicas divulgadas. Como un ejemplo, las técnicas pueden permitir que el descodificador de vídeo descodifique el flujo de bits con una granularidad relativamente mayor. En otras palabras, las técnicas pueden permitir al descodificador de vídeo acceder al flujo de bits aleatoriamente en relativamente más puntos, o imágenes (es decir, imágenes no IDR) del flujo de bits, en comparación con otras técnicas (por ejemplo, técnicas que permiten el acceso aleatorio de un flujo de bits solo de imágenes IDR). Como otro ejemplo, puede haber una mejora relativa en la calidad visual de una o más imágenes de la CVS incluidas en el flujo de bits, y/o de la CVS como un todo (por ejemplo, mediante el codificador de vídeo 20 omitiendo las imágenes principales asociadas con el primer imagen del flujo de bits), cuando se usan las técnicas divulgadas.

[0176] De esta manera, el codificador de vídeo 20 representa un ejemplo de un codificador de vídeo configurado para generar un flujo de bits que incluye una o más imágenes de una CVS, en el que una primera imagen de la una o más imágenes de acuerdo con un orden de descodificación asociado con la CVS es una imagen RAP que no es una imagen IDR, en el que para generar el flujo de bits, el codificador de vídeo está configurado para evitar

incluir al menos una de las una o más imágenes, distintas de la primera imagen, que corresponde a una imagen principal asociada con la primera imagen en el flujo de bits, en el que la imagen principal es una imagen que sigue a la primera imagen de acuerdo con el orden de descodificación y precede a la primera imagen de acuerdo con un orden de visualización asociado con la CVS, y en el que la primera imagen es descodificable, y en el que al menos uno de la una o más imágenes, distintas de la primera imagen, que siguen a la primera imagen de acuerdo con el orden de descodificación, se pueden descodificar basándose en la primera imagen.

[0177] La FIG. 3 es un diagrama de bloques que ilustra un ejemplo de un descodificador de vídeo que puede implementar las técnicas para acceso aleatorio con gestión de DPB avanzada, de forma coherente con las técnicas de esta divulgación. En el ejemplo de la FIG. 3, el descodificador de vídeo 30 incluye una unidad de descodificación por entropía 80, una unidad de procesamiento de predicción 82, una unidad de cuantificación inversa 88, una unidad de procesamiento de transformación inversa 90, un sumador 92, un filtro de desbloqueo 94 y una memoria de imágenes de referencia 96. La unidad de procesamiento de predicción 82 incluye la unidad de compensación de movimiento 84 y la unidad de procesamiento de predicción intra 86. En algunos ejemplos, el descodificador de vídeo 30 puede realizar 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.

[0178] Durante el proceso 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. Cuando los bloques de vídeo representados en el flujo de bits incluyen datos de vídeo comprimidos, la unidad de descodificación por entropía 80 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 80 remite los vectores de movimiento y otros elementos sintácticos a la unidad de procesamiento de predicción 82. El descodificador de vídeo 30 puede recibir los elementos sintácticos en el nivel del fragmento de vídeo y/o el nivel del bloque de vídeo.

[0179] Cuando el fragmento de vídeo se codifica como un fragmento intra-codificado (I), la unidad de procesamiento de predicción intra 86 de la unidad de procesamiento de predicción 82 puede generar datos de predicción para un bloque de vídeo del fragmento de vídeo actual, basándose en un modo de predicción intra señalado, y datos de bloques previamente descodificados de la trama o imagen actual. Cuando la trama de vídeo es codificada como un fragmento inter-codificado (es decir, B, P o GPB), la unidad de compensación de movimiento 84 de la unidad de procesamiento de predicción 82 genera 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 de entropía 80. Los bloques predictivos pueden generarse 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 tramas 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 96.

[0180] La unidad de compensación de movimiento 84 determina la información de predicción para un bloque de vídeo del fragmento de vídeo actual analizando los vectores de movimiento y otros elementos sintácticos, y usa la información de predicción para generar los bloques predictivos del bloque de vídeo actual que se está descodificando. Por ejemplo, la unidad de compensación de movimiento 84 usa algunos de los elementos sintácticos recibidos para determinar un modo de predicción (por ejemplo, predicción intra o predicción inter) usado para codificar los bloques de vídeo del fragmento de vídeo, un tipo de fragmento de predicción inter (por ejemplo, fragmento B, fragmento P o fragmento GPB), información de construcción para una o más de las listas de imágenes de referencia del fragmento, vectores de movimiento para cada bloque de vídeo inter-codificado del fragmento, el estado de predicción inter para cada bloque de vídeo inter-codificado del fragmento y otra información para descodificar los bloques de vídeo en el fragmento de vídeo actual.

[0181] La unidad de compensación de movimiento 84 también puede realizar la interpolación basándose en filtros de interpolación. La unidad de compensación de movimiento 84 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 para píxeles fraccionarios de bloques de referencia. La unidad de compensación de movimiento 84 puede determinar los filtros de interpolación utilizados por el codificador de vídeo 20 a partir de los elementos sintácticos recibidos, y utilizar los filtros de interpolación para producir bloques predictivos.

[0182] La unidad de cuantificación inversa 88 cuantifica de manera inversa, es decir, descuantifica, los coeficientes de transformación cuantificados proporcionados en el flujo de bits y descodificados por la unidad de descodificación por entropía 80. 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 en el fragmento de vídeo, para determinar un grado de cuantificación y, asimismo, un grado de cuantificación inversa que debería aplicarse. La unidad de procesamiento de transformación inversa 90 aplica una transformación inversa, por ejemplo una DCT inversa, una transformación inversa entera o un proceso de transformación inversa conceptualmente similar, a los coeficientes de transformación, con el fin de generar bloques residuales en el dominio de píxeles.

[0183] Después de que la unidad de compensación de movimiento 84 genera el bloque predictivo para el bloque de vídeo actual, basándose en los vectores de movimiento y 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 transformación inversa 90 con los correspondientes bloques predictivos generados por la unidad de compensación de movimiento 84. El sumador 92 representa el componente o los componentes que llevan a cabo esta operación de suma. Se aplica un filtro de eliminación de bloques 94 para filtrar los bloques descodificados, con el fin de eliminar distorsiones de efecto pixelado. Los bloques de vídeo descodificados en una trama o imagen dada son a continuación almacenados en la memoria de imágenes de referencia 96, que almacena imágenes de referencia usadas para la posterior compensación de movimiento. La memoria de imágenes de referencia 96 almacena también vídeo descodificado para su presentación posterior en un dispositivo de visualización, tal como el dispositivo de visualización 28 de la FIG. 1.

[0184] Como un ejemplo, el descodificador de vídeo 30 puede configurarse para descodificar una o más imágenes de datos de vídeo durante un proceso de descodificación de vídeo. Por ejemplo, el descodificador de vídeo 30 puede configurarse para recibir un flujo de bits, por ejemplo, generado por el codificador de vídeo 20, que incluye una o más imágenes de una CVS. El descodificador de vídeo 30 puede configurarse adicionalmente para descodificar una primera imagen de la una o más imágenes de acuerdo con un orden de descodificación asociado con la CVS. En este ejemplo, la primera imagen puede ser una imagen RAP que no es una imagen IDR. El descodificador de vídeo 30 también puede configurarse para descodificar al menos una de las una o más imágenes, distintas de la primera imagen, siguiendo a la primera imagen de acuerdo con el orden de descodificación, basándose en la primera imagen descodificada.

[0185] En algunos ejemplos, el descodificador de vídeo 30 puede configurarse adicionalmente para determinar (o identificar) al menos una de las una o más imágenes, distintas de la primera imagen, que corresponde a una imagen principal asociada con la primera imagen. En otras palabras, el descodificador de vídeo 30 puede configurarse para identificar al menos una imagen principal que está asociada con la primera imagen entre la una o más imágenes. En estos ejemplos, la imagen principal puede ser una vez más una imagen que sigue a la primera imagen de acuerdo con el orden de descodificación y precede a la primera imagen de acuerdo con un orden de visualización asociado con la CVS. Por ejemplo, el descodificador de vídeo 30 puede configurarse para descodificar la al menos una de las una o más imágenes. Para descodificar cada una de las al menos una de las una o más imágenes, el descodificador de vídeo 30 puede configurarse para determinar, o identificar, una o más imágenes de referencia usadas para codificar la imagen respectiva, determinar si alguna de las una o más imágenes de referencia determinadas o identificadas no están disponibles para descodificarse, para cada una de las una o más imágenes de referencia determinadas o identificadas que se determina que no están disponibles para ser descodificadas, generar una imagen de referencia virtual y descodificar la imagen respectiva basándose en la una o más imágenes de referencia virtuales generadas.

[0186] En los ejemplos descritos anteriormente, para generar la imagen de referencia virtual, el descodificador de vídeo 30 puede configurarse para generar una imagen que incluye uno o más valores de píxel, cada uno de los cuales corresponde a un valor medio de un rango de valores de píxeles asociados con la CVS. Por ejemplo, el descodificador de vídeo 30 puede configurarse para generar la imagen de modo que la imagen incluya uno o más valores de píxel, cada uno de los cuales teniendo valores "luma" o "croma" de "127". En este ejemplo, cada valor de píxel puede corresponder a un valor medio de un rango de valores de intensidad de píxel de luma o croma definidos a partir de un valor de intensidad de píxel de "0" a un valor de intensidad de píxel de "255". Por ejemplo, cada uno de los valores de intensidad de píxel de luma o croma se puede representar usando 7 bits de datos, lo cual da como resultado el rango de valores descrito anteriormente. En otros ejemplos, sin embargo, el rango de intensidad de píxel de luma o croma, y los valores medios o medios correspondientes del rango, se pueden definir de una manera diferente.

[0187] En algunos ejemplos, el descodificador de vídeo 30 puede configurarse adicionalmente para determinar, o identificar, al menos una de las una o más imágenes, distintas de la primera imagen, que corresponde a una imagen principal asociada con la primera imagen. En estos ejemplos, una vez más, la imagen principal puede ser una imagen que sigue a la primera imagen de acuerdo con el orden de descodificación y precede a la primera imagen de acuerdo con un orden de visualización asociado con la CVS. También en estos ejemplos, el descodificador de vídeo 30 puede configurarse para no emitir, o evitar emitir, una o más de al menos una de las una o más imágenes (es decir, una o más de las imágenes principales previamente determinadas o identificadas asociadas con la primera imagen) para el cual un indicador de salida (por ejemplo, el indicador de salida del elemento sintáctico descrito anteriormente) indica que la imagen respectiva se va a emitir.

[0188] En otros ejemplos, el descodificador de vídeo 30 puede configurarse aún más para determinar, o identificar, al menos una de las una o más imágenes, distintas de la primera imagen, que corresponde a una imagen principal asociada con la primera imagen. En estos ejemplos, una vez más, la imagen principal puede ser una imagen que sigue a la primera imagen de acuerdo con el orden de descodificación y precede a la primera imagen de acuerdo con un orden de visualización asociado con la CVS. También en estos ejemplos, el descodificador de vídeo 30 puede configurarse para no usar, o evitar el uso, de una o más de al menos una de las una o más imágenes (es decir, una o más de las imágenes principales previamente determinadas o identificadas

asociadas con la primera imagen) como una imagen de referencia para descodificar al menos una de las una o más imágenes, distintas de la primera imagen, que sigue a la primera imagen de acuerdo con el orden de descodificación y de acuerdo con un orden de visualización asociado con la CVS.

5 **[0189]** En algunos ejemplos, la primera imagen puede ser una imagen de CRA. En estos ejemplos, la imagen de CRA puede ser una imagen que se codifica utilizando la codificación de predicción intra y se puede descodificar, es decir, se puede descodificar, sin referencia a ninguna otra imagen. En estos ejemplos, para la imagen de CRA, una o más imágenes incluidas en una CVS junto con la imagen de CRA que siguen a la imagen de CRA de acuerdo con un orden de descodificación asociado con la CVS pueden descodificarse con referencia a una o más imágenes que preceden a la imagen de CRA de acuerdo con el orden de descodificación. Por ejemplo, la imagen de CRA puede referirse a una imagen intra-codificada de "Open-GOP", como se describió anteriormente con referencia a la FIG. 1. Como también se describió anteriormente, la imagen de CRA puede tener un propósito similar a una imagen IDR en un ajuste de "Closed-GOP", particularmente con respecto a habilitar el acceso aleatorio de un flujo de bits que incluye una o más imágenes de uno o más GOP de datos de vídeo.

15 **[0190]** En otros ejemplos, la imagen IDR puede ser una imagen que se codifica utilizando codificación de predicción intra y puede descodificarse, es decir, es descodificable, sin referencia a ninguna otra imagen. En estos ejemplos, para la imagen IDR, todas las demás imágenes incluidas en una CVS junto con la imagen IDR que sigue a la imagen IDR de acuerdo con un orden de descodificación asociado con la CVS pueden descodificarse sin referencia a ninguna imagen que preceda a la imagen IDR de acuerdo con el orden de descodificación.

20 **[0191]** En otros ejemplos, por ejemplo, en casos donde el flujo de bits recibido por el descodificador de vídeo 30 no incluye ninguna imagen principal asociada con la primera imagen (por ejemplo, en casos donde el codificador de vídeo 20 generó el flujo de bits excluyendo las imágenes principales de la primera imagen del flujo de bits), el descodificador de vídeo 30 puede configurarse para descodificar el flujo de bits de una manera particular, como se ilustra por los ejemplos a continuación.

25 **[0192]** Como un ejemplo, el descodificador de vídeo 30 puede configurarse adicionalmente para descodificar un primer conjunto de parámetros de retardo inicial de memoria intermedia de imágenes codificadas (CPB) y, cuando la una o más imágenes no incluyen al menos una imagen principal asociada con la primera imagen, descodificar uno de un segundo conjunto de parámetros de retardo inicial de CPB, en el que el segundo conjunto es diferente del primer conjunto, y un conjunto de parámetros de compensación de retardo de CPB. En este ejemplo, una vez más, la imagen principal puede ser una imagen que sigue a la primera imagen de acuerdo con el orden de descodificación y precede a la primera imagen de acuerdo con un orden de visualización asociado con la CVS.

30 **[0193]** En el ejemplo descrito anteriormente, uno o más de los primeros y segundos conjuntos de parámetros de retardo inicial de CPB y el conjunto de parámetros de compensación de retardo de CPB pueden incluirse en uno de un mensaje de información de mejora suplementaria (SEI), un mensaje SEI de período de memoria intermedia de imágenes y una cabecera de fragmento, asociados a la primera imagen.

35 **[0194]** También en el ejemplo descrito anteriormente, un momento de eliminación de CPB de cada imagen después de la primera imagen en el orden de descodificación puede cambiarse antes según lo indicado por uno o más de los primeros y segundos conjuntos de parámetros de retardo inicial de CPB y el conjunto de retardo de CPB parámetros de compensación de retardo.

40 **[0195]** Por consiguiente, como se explicó anteriormente, las técnicas de esta divulgación pueden permitir que el descodificador de vídeo 30 descodifique un flujo de bits, por ejemplo, codificado por el codificador de vídeo 20, de una manera predecible y definida, como se especifica mediante las técnicas de esta divulgación. En particular, el flujo de bits puede incluir una o más imágenes de una CVS de datos de vídeo. El flujo de bits puede ser recibido por el descodificador de vídeo 30 de manera que el flujo de bits comience con una imagen RAP no IDR. Usando las técnicas de esta divulgación, el descodificador de vídeo 30 puede descodificar con éxito el flujo de bits. Como tal, puede haber una mejora relativa en la experiencia del usuario cuando se realiza un acceso aleatorio del flujo de bits, cuando se usan las técnicas divulgadas. Como un ejemplo, las técnicas pueden permitir que el descodificador de vídeo 30 descodifique el flujo de bits con una granularidad relativamente mayor. Dicho de otra manera, las técnicas pueden permitir que el descodificador de vídeo 30 acceda aleatoriamente al flujo de bits en relativamente más puntos, o imágenes (es decir, imágenes no IDR) del flujo de bits, en comparación con otras técnicas (por ejemplo, técnicas que permiten el acceso aleatorio de un flujo de bits solo de imágenes IDR). Como otro ejemplo, puede haber una mejora relativa en la calidad visual de una o más imágenes de la CVS incluidas en el flujo de bits, y/o de la CVS como un todo (por ejemplo, mediante un descodificador de vídeo 30 que evita emitir y usar como imágenes de referencia las imágenes principales asociadas con la primera imagen), cuando se usan las técnicas divulgadas.

45 **[0196]** De esta manera, el descodificador de vídeo 30 representa un ejemplo de un descodificador de vídeo configurado para recibir un flujo de bits que comprende una o más imágenes de una CVS, descodificar una primera imagen de la una o más imágenes de acuerdo con un orden de descodificación asociado con la CVS, en el que la primera imagen es una imagen RAP que no es una imagen IDR, y descodificar al menos una de las una o más

imágenes, distintas de la primera imagen, siguiendo la primera imagen de acuerdo con el orden de descodificación, basándose en la primera imagen descodificada.

[0197] Las FIGs. 5 y 6 son diagramas de flujo que ilustran procedimientos de ejemplo para realizar las técnicas de acceso aleatorio con gestión avanzada de DPB, de acuerdo con las técnicas de esta divulgación. En particular, el procedimiento de ejemplo de la FIG. 5 ilustra la realización de las técnicas desde el punto de vista de un descodificador de vídeo, por ejemplo, el descodificador de vídeo 30 de las FIGs. 1 y 3. Además, el procedimiento de ejemplo de la FIG. 6 ilustra el modo de realización de las técnicas desde el punto de vista de un codificador de vídeo, por ejemplo, el codificador de vídeo 20 de las FIGs. 1 y 2.

[0198] Las técnicas de las FIGs. 5 y 6 en general pueden realizarse mediante cualquier unidad de procesamiento o procesador, ya sea implementado en hardware, software, firmware o una combinación de los mismos, y cuando se implementa en software o firmware, se puede proporcionar el hardware correspondiente para ejecutar instrucciones para el software o firmware. Como ejemplo, las técnicas de las FIGs. 5 y 6 se describen con respecto a varios componentes del codificador de vídeo 20 (FIG. 2), y/o el descodificador de vídeo 30 (FIG. 3) aunque se debe entender que otros dispositivos pueden configurarse para realizar técnicas similares. Además, los pasos ilustrados en las FIGs. 5 y 6 pueden realizarse en un orden diferente o en paralelo, y pueden agregarse pasos adicionales y omitirse ciertos pasos, sin apartarse de las técnicas de esta divulgación. Además, de acuerdo con las técnicas de esta divulgación, las técnicas de los procedimientos de ejemplo de las FIGs. 5 y 6 pueden realizarse individualmente o en combinación una con la otra.

[0199] La FIG. 5 es un diagrama de flujo que ilustra un procedimiento de ejemplo para realizar un acceso aleatorio de un flujo de bits que incluye una o más imágenes de datos de vídeo mediante un descodificador de vídeo, por ejemplo, un descodificador de vídeo 30 de las FIGs. 1 y 3, de forma coherente con las técnicas de esta divulgación. En particular, las técnicas del procedimiento de ejemplo de la FIG. 5 incluyen realizar el acceso aleatorio del flujo de bits en casos en los que una primera imagen del flujo de bits es una imagen RAP no IDR de una manera específica, como se describe a continuación.

[0200] Como un ejemplo, el descodificador de vídeo 30 puede recibir un flujo de bits que incluye una o más imágenes de una CVS (500). El descodificador de vídeo 30 puede descodificar adicionalmente una primera imagen de la una o más imágenes de acuerdo con un orden de descodificación asociado con la CVS, en el que la primera imagen es una imagen RAP que no es una imagen IDR (502). El descodificador de vídeo 30 también puede descodificar al menos una de las una o más imágenes, distintas de la primera imagen, siguiendo a la primera imagen de acuerdo con el orden de descodificación, basándose en la primera imagen descodificada (504).

[0201] En algunos ejemplos, el descodificador de vídeo 30 puede además determinar, o identificar, al menos una de las una o más imágenes, distintas de la primera imagen, que corresponde a una imagen principal asociada con la primera imagen. En estos ejemplos, una vez más, la imagen principal puede ser una imagen que sigue a la primera imagen de acuerdo con el orden de descodificación y precede a la primera imagen de acuerdo con un orden de visualización asociado con la CVS. El descodificador de vídeo 30 puede descodificar adicionalmente la al menos una de las una o más imágenes. En estos ejemplos, para descodificar cada una de las al menos una de las una o más imágenes, el descodificador de vídeo 30 puede realizar los siguientes pasos: (1) determinar, o identificar, una o más imágenes de referencia utilizadas para codificar la imagen respectiva; (2) determinar si alguna de las una o más imágenes de referencia determinadas o identificadas no está disponible para descodificarse; (3) para cada una de las una o más imágenes de referencia determinadas o identificadas que se determina que no están disponibles para descodificarse, generar una imagen de referencia virtual; y (4) descodificar la imagen respectiva basándose en las correspondientes una o más imágenes de referencia virtuales generadas.

[0202] En los ejemplos descritos anteriormente, para generar la imagen de referencia virtual, el descodificador de vídeo 30 puede generar una imagen que incluye uno o más valores de píxel, cada uno de los cuales corresponde a un valor medio de un rango de valores de píxeles asociados con la CVS (por ejemplo, uno o más valores de píxel de luma o croma de "127" dentro de un rango de "0" a "255"), como se describió anteriormente con referencia a la FIG. 3.

[0203] En algunos ejemplos, el descodificador de vídeo 30 puede además determinar, o identificar, al menos una de las una o más imágenes, distintas de la primera imagen, que corresponde a una imagen principal asociada con la primera imagen. En estos ejemplos, una vez más, la imagen principal puede ser una imagen que sigue a la primera imagen de acuerdo con el orden de descodificación y precede a la primera imagen de acuerdo con un orden de visualización asociado con la CVS. También en estos ejemplos, el descodificador de vídeo 30 puede evitar adicionalmente emitir una o más de al menos una de las una o más imágenes (es decir, una o más de las imágenes principales determinadas o identificadas previamente asociadas con la primera imagen) para las cuales un indicador de salida (por ejemplo, indicador de salida del elemento sintáctico) indica que la imagen respectiva se va a emitir.

[0204] En otros ejemplos, el descodificador de vídeo 30 puede además determinar, o identificar, al menos una de las una o más imágenes, distintas de la primera imagen, que corresponde a una imagen principal asociada con la primera imagen. En estos ejemplos, una vez más, la imagen principal puede ser una imagen que sigue a la primera imagen de acuerdo con el orden de descodificación y precede a la primera imagen de acuerdo con un orden de visualización asociado con la CVS. También en estos ejemplos, el descodificador de vídeo 30 puede evitar adicionalmente el uso de una o más de al menos una de las una o más imágenes (es decir, una o más de las imágenes principales determinadas o identificadas previamente asociadas con la primera imagen) como una imagen de referencia para descodificar al menos una de las una o más imágenes, distintas de la primera imagen, que sigue a la primera imagen de acuerdo con el orden de descodificación y de acuerdo con un orden de visualización asociado con la CVS.

[0205] En los ejemplos descritos anteriormente, la primera imagen puede ser una imagen de CRA. En estos ejemplos, la imagen de CRA puede ser una imagen que se codifica utilizando la codificación de predicción intra y se puede descodificar, es decir, se puede descodificar, sin referencia a ninguna otra imagen. En estos ejemplos, para la imagen de CRA, una o más imágenes incluidas en una CVS junto con la imagen de CRA que siguen a la imagen de CRA de acuerdo con un orden de descodificación asociado con la CVS pueden descodificarse con referencia a una o más imágenes que preceden a la imagen de CRA de acuerdo con el orden de descodificación. Por ejemplo, como se describió anteriormente, la imagen de CRA puede referirse a una imagen intra-codificada de "Open-GOP", como se describió anteriormente con referencia a la FIG. 1. Como también se describió anteriormente, la imagen de CRA puede tener un propósito similar a una imagen IDR en un ajuste de "Closed-GOP", particularmente con respecto a habilitar el acceso aleatorio de un flujo de bits que incluye una o más imágenes de uno o más GOP de datos de vídeo.

[0206] También en los ejemplos descritos anteriormente, la imagen IDR puede ser una imagen que se codifica utilizando codificación de predicción intra y se puede descodificar, es decir, es descodificable, sin referencia a ninguna otra imagen. Además, la imagen IDR puede ser una imagen para la cual todas las demás imágenes incluidas en una CVS junto con la imagen IDR que sigue a la imagen IDR de acuerdo con un orden de descodificación asociado con la CVS son descodificadas sin referencia a ninguna imagen que preceda a la imagen IDR de acuerdo con el orden de descodificación.

[0207] En otros ejemplos, por ejemplo, en casos donde el flujo de bits recibido por el descodificador de vídeo 30 no incluye ninguna imagen principal asociada con la primera imagen (por ejemplo, en casos donde el codificador de vídeo 20 generó el flujo de bits excluyendo las imágenes principales de la primera imagen del flujo de bits), el descodificador de vídeo 30 puede descodificar el flujo de bits de una manera particular, como se ilustra mediante los ejemplos siguientes.

[0208] Como ejemplo, el descodificador de vídeo 30 puede descodificar adicionalmente un primer conjunto de parámetros de retardo inicial de CPB, y, cuando la una o más imágenes no incluyen al menos una imagen principal asociada con la primera imagen, descodificar uno de un segundo conjunto de parámetros de retardo inicial de CPB, en los que el segundo conjunto es diferente del primer conjunto, y un conjunto de parámetros de compensación de retardo de CPB. En este ejemplo, una vez más, la imagen principal puede ser una imagen que sigue a la primera imagen de acuerdo con el orden de descodificación y precede a la primera imagen de acuerdo con un orden de visualización asociado con la CVS.

[0209] En el ejemplo descrito anteriormente, uno o más de los primeros y segundos conjuntos de parámetros de retardo inicial de CPB y el conjunto de parámetros de compensación de retardo de CPB pueden incluirse en uno de un mensaje SEI, un mensaje SEI de período de almacenamiento en memoria intermedia de imágenes y una cabecera de sector, asociado con la primera imagen.

[0210] También en el ejemplo descrito anteriormente, un momento de eliminación de CPB de cada imagen después de la primera imagen en el orden de descodificación puede cambiarse antes según lo indicado por uno o más de los primeros y segundos conjuntos de parámetros de retardo inicial de CPB y el conjunto de retardo de CPB parámetros de compensación de retardo.

[0211] De esta manera, el procedimiento de la FIG. 5 representa un ejemplo de un procedimiento de recepción de un flujo de bits que comprende una o más imágenes de una CVS, descodificar una primera imagen de la una o más imágenes de acuerdo con un orden de descodificación asociado con la CVS, en el que la primera imagen es una imagen RAP que es no es una imagen IDR, y descodificar al menos una de las una o más imágenes, aparte de la primera imagen, que sigue a la primera imagen de acuerdo con el orden de descodificación, basándose en la primera imagen descodificada.

[0212] La FIG. 6 es un diagrama de flujo que ilustra un procedimiento de ejemplo para generar un flujo de bits que incluye una o más imágenes de datos de vídeo mediante un codificador de vídeo, por ejemplo, el codificador de vídeo 20 de las FIGs. 1 y 2, de acuerdo con las técnicas de esta divulgación. En particular, las técnicas del procedimiento de ejemplo de la FIG. 6 incluyen generar el flujo de bits, de modo que un descodificador de vídeo, por ejemplo, descodificador de vídeo 30, pueda descodificar con éxito el flujo de bits de una manera específica.

Por ejemplo, el descodificador de vídeo puede descodificar el flujo de bits en los casos en que una primera imagen del flujo de bits es una imagen RAP no IDR, como se describe a continuación.

5 **[0213]** Como un ejemplo, el codificador de vídeo 20 puede generar un flujo de bits que incluye una o más imágenes de una CVS. En este ejemplo, una primera imagen de la una o más imágenes de acuerdo con un orden de descodificación asociado con la CVS puede ser una imagen RAP que no es una imagen IDR (600). El codificador de vídeo 20 puede evitar adicionalmente incluir al menos una de las una o más imágenes, distintas de la primera imagen, que corresponde a una imagen principal asociada con la primera imagen en el flujo de bits. En este ejemplo, una vez más, la imagen principal puede ser una imagen que sigue a la primera imagen de acuerdo con el orden de descodificación y precede a la primera imagen de acuerdo con un orden de visualización asociado con la CVS (602).

15 **[0214]** En este ejemplo, posteriormente, un descodificador de vídeo, por ejemplo, un descodificador de vídeo 30, puede recibir el flujo de bits generado por el codificador de vídeo 20 y descodificar el flujo de bits. Por ejemplo, el descodificador de vídeo puede descodificar la primera imagen. El descodificador de vídeo puede descodificar adicionalmente al menos una de las una o más imágenes, distintas de la primera imagen, siguiendo a la primera imagen de acuerdo con el orden de descodificación, basándose en la primera imagen (por ejemplo, basándose en una versión descodificada de la primera imagen).

20 **[0215]** De esta manera, el procedimiento de la FIG. 6 representa un ejemplo de un procedimiento para generar un flujo de bits que comprende una o más imágenes de una CVS, en el que una primera imagen de la una o más imágenes de acuerdo con un orden de descodificación asociado con la CVS es una imagen RAP que no es una imagen IDR, en el que generar el flujo de bits comprende evitar incluir al menos una de las una o más imágenes, distintas de la primera imagen, que corresponde a una imagen principal asociada con la primera imagen en el flujo de bits, en el que la imagen principal comprende una imagen que sigue a la primera imagen de acuerdo con el orden de descodificación y precede a la primera imagen de acuerdo con un orden de visualización asociado con la CVS, y en el que la primera imagen es descodificable, y en el que al menos una de las una o más imágenes, aparte de la primera imagen, después de la primera imagen de acuerdo con el orden de descodificación, es descodificable basándose en la primera imagen.

30 **[0216]** En uno o más ejemplos, las funciones descritas en el presente documento pueden implementarse en hardware, software, firmware o cualquier combinación de estos. Si se implementan en software, las funciones pueden almacenarse en o transmitirse a través de, como una o más instrucciones o código, un medio legible por ordenador o ejecutarse mediante una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que pueden corresponder a un medio tangible o no transitorio tal como unos medios de almacenamiento de datos o unos medios de comunicación que incluyen cualquier medio que facilite la transferencia de un programa informático desde 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) unos 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 medios disponibles cualesquiera a los 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.

45 **[0217]** A modo de ejemplo, y no de manera limitativa, tales medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento de disco óptico, almacenamiento de disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que pueda usarse para almacenar código de programa deseado en forma de instrucciones o estructuras de datos y al que pueda accederse mediante un ordenador. Además, cualquier conexión recibe debidamente 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 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 se incluyen en la definición de medio. Sin embargo, debería entenderse que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales ni otros medios transitorios, sino que, en cambio, se orientan a medios de almacenamiento tangibles estacionarios o no transitorios. El término disco, tal como se utiliza en el presente documento, incluye un disco compacto (CD), un disco láser, un disco óptico, un disco versátil digital (DVD), un disco flexible y un disco Blu-ray, donde algunos discos habitualmente emiten datos magnéticamente, mientras que otros discos emiten datos ópticamente con láseres. Las combinaciones de los anteriores también deben incluirse dentro del alcance de los medios legibles por ordenador.

65 **[0218]** Las instrucciones pueden ejecutarse mediante uno o más procesadores, tales como uno o más microprocesadores de uso general, DSP, ASIC, FPGA u otra circuitería lógica integrada o discreta equivalente. Por consiguiente, el término "procesador", como se utiliza en el presente documento, puede referirse a cualquiera

de las estructuras anteriores o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en esta divulgación. Además, en algunos aspectos, la funcionalidad descrita en el presente documento puede proporcionarse dentro de módulos de hardware y/o software dedicados configurados para la codificación y la descodificación, o incorporarse en un códec combinado. Asimismo, las técnicas podrían implementarse por completo en uno o más circuitos o elementos lógicos.

5

[0219] Las técnicas de esta divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, incluidos un teléfono inalámbrico, un CI o un conjunto de CI (por ejemplo, un conjunto de chips). En esta divulgación se describen diversos componentes, módulos o unidades para destacar los aspectos funcionales de los dispositivos configurados para realizar las técnicas divulgadas, pero no requieren necesariamente la realización mediante diferentes componentes, módulos o unidades de hardware. En cambio, como se ha descrito anteriormente, diversas unidades pueden combinarse en una unidad de hardware de códec o proporcionarse por medio de un grupo de unidades de hardware interoperativas, que incluyen uno o más procesadores como los descritos anteriormente, conjuntamente con software y/o firmware adecuados.

10

15

[0220] 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 (500) un flujo de bits que comprende una o más secuencias de vídeo codificadas, CVS, cada CVS que comprende una o más imágenes, en el que una primera imagen en orden de descodificación en una primera CVS es una imagen de punto de acceso aleatorio, RAP, que no es una imagen de actualización de descodificación instantánea, IDR;

10 descodificar un primer conjunto de parámetros de retardo inicial de memoria intermedia de imágenes codificadas, CPB;

15 descodificar, cuando la una o más imágenes de una CVS no incluyen al menos una imagen principal asociada con la primera imagen, un segundo conjunto de parámetros de retardo inicial de CPB, en el que el segundo conjunto de parámetros de retardo inicial de CPB es diferente del primer conjunto de parámetros de retardo inicial de CPB; y

20 seleccionar uno de los dos conjuntos de parámetros de retardo inicial de CPB para una obtención de los parámetros de temporización de CPB,

25 en el que, cuando una o más imágenes de la CVS no incluyen al menos una imagen principal asociada con la primera imagen, los parámetros de temporización de la CPB se obtienen basándose en el segundo conjunto de parámetros de retardo inicial de CPB y un momento de eliminación de CPB de cada imagen siguiente a la primera imagen en el orden de descodificación se desplaza antes, como lo indica mediante el segundo conjunto de parámetros de retardo inicial de CPB,

30 en el que la imagen principal comprende una imagen que sigue a la primera imagen de acuerdo con el orden de descodificación y precede a la primera imagen de acuerdo con un orden de visualización asociado con la CVS.

2. El procedimiento según la reivindicación 1, en el que los conjuntos primero y segundo de parámetros de retardo inicial de CPB se incluyen en uno de un mensaje de información de mejora suplementaria, SEI, un mensaje SEI de período de memoria intermedia de imágenes y una cabecera de fragmento, asociado con la primera imagen.

3. El procedimiento según la reivindicación 1 o la reivindicación 2, en el que la primera imagen comprende una imagen de acceso aleatorio limpio, CRA, en el que la imagen de CRA comprende una imagen que se codifica usando codificación de predicción intra y se puede descodificar sin referencia a otras imágenes, y para el que una o más imágenes incluidas dentro de una CVS junto con la imagen de CRA que siguen a la imagen de CRA de acuerdo con un orden de descodificación asociado con la CVS pueden descodificarse con referencia a una o más imágenes que preceden a la imagen de CRA de acuerdo con el orden de descodificación.

4. El procedimiento de cualquiera de las reivindicaciones 1 a 3, en el que la imagen IDR comprende una imagen que se codifica usando codificación de predicción intra y es descodificable sin referencia a ninguna otra imagen, y para la cual todas las demás imágenes incluidas dentro de una CVS junto con la imagen IDR que siguen a la imagen IDR de acuerdo con un orden de descodificación asociado con la CVS se descodifican sin referencia a ninguna imagen que preceda a la imagen IDR de acuerdo con el orden de descodificación.

5. Un dispositivo (14) para descodificar datos de vídeo, el dispositivo que comprende:

55 medios para recibir un flujo de bits que comprende una o más secuencias de vídeo codificadas, CVS, cada CVS que comprende una o más imágenes, en el que una primera imagen en orden de descodificación es una imagen de punto de acceso aleatorio, RAP, que no es una imagen de actualización de descodificación instantánea, IDR; y

60 medios para descodificar un primer conjunto de parámetros de retardo inicial de memoria intermedia de imágenes codificadas, CPB;

65 medios para descodificar, cuando la una o más imágenes de una CVS no incluyen al menos una imagen principal asociada con la primera imagen, un segundo conjunto de parámetros de retardo inicial de CPB, en el que el segundo conjunto de parámetros de retardo inicial de CPB es diferente al primero conjunto de parámetros de retardo inicial de CPB; y

medios para seleccionar uno de los dos conjuntos de parámetros de retardo inicial de CPB para una obtención de parámetros de temporización de CPB,

5 en el que, cuando una o más imágenes de la CVS no incluyen al menos una imagen principal asociada con la primera imagen, los parámetros de temporización de la CPB se obtienen basándose en el segundo conjunto de parámetros de retardo inicial de CPB y un momento de eliminación de CPB de cada imagen siguiente a la primera imagen en el orden de descodificación se desplaza antes, como lo indica mediante el segundo conjunto de parámetros de retardo inicial de CPB,

10 en el que la imagen principal comprende una imagen que sigue a la primera imagen de acuerdo con el orden de descodificación y precede a la primera imagen de acuerdo con un orden de visualización asociado con la CVS.

15 **6.** El dispositivo de la reivindicación 5, en el que los primer y el segundo conjuntos de parámetros de retardo inicial de CPB están incluidos en uno de un mensaje de información de mejora complementaria, SEI, un mensaje SEI de período de almacenamiento en memoria intermedia de imágenes y una cabecera de fragmento, asociado a la primera imagen.

20 **7.** El dispositivo de la reivindicación 5 o la reivindicación 6, en el que la primera imagen comprende una imagen de acceso aleatorio limpio, CRA, en el que la imagen de CRA comprende una imagen codificada mediante codificación de predicción inter y puede descodificarse sin referencia a otras imágenes, y para el que una o más imágenes incluidas dentro de una CVS junto con la imagen de CRA que siguen a la imagen de CRA de acuerdo con un orden de descodificación asociado con la CVS pueden descodificarse con referencia a una o más imágenes que preceden a la imagen de CRA de acuerdo con el orden de descodificación.

25 **8.** El dispositivo de cualquiera de las reivindicaciones 5 a 7, en el que la imagen IDR comprende una imagen que se codifica usando codificación de predicción intra y se puede descodificar sin referencia a ninguna otra imagen, y para el cual todas las demás imágenes incluidas dentro de una CVS junto con la imagen IDR que siguen a la imagen IDR de acuerdo con un orden de descodificación asociado con la CVS se descodifican sin referencia a ninguna imagen que preceda a la imagen IDR de acuerdo con el orden de descodificación.

30 **9.** Un medio de almacenamiento legible por ordenador que almacena instrucciones que, cuando se ejecutan, hacen que uno o más procesadores descodifiquen datos de vídeo, en el que las instrucciones hacen que el uno o más procesadores lleven a cabo un procedimiento de acuerdo con cualquiera de las reivindicaciones 1 a 4.

35

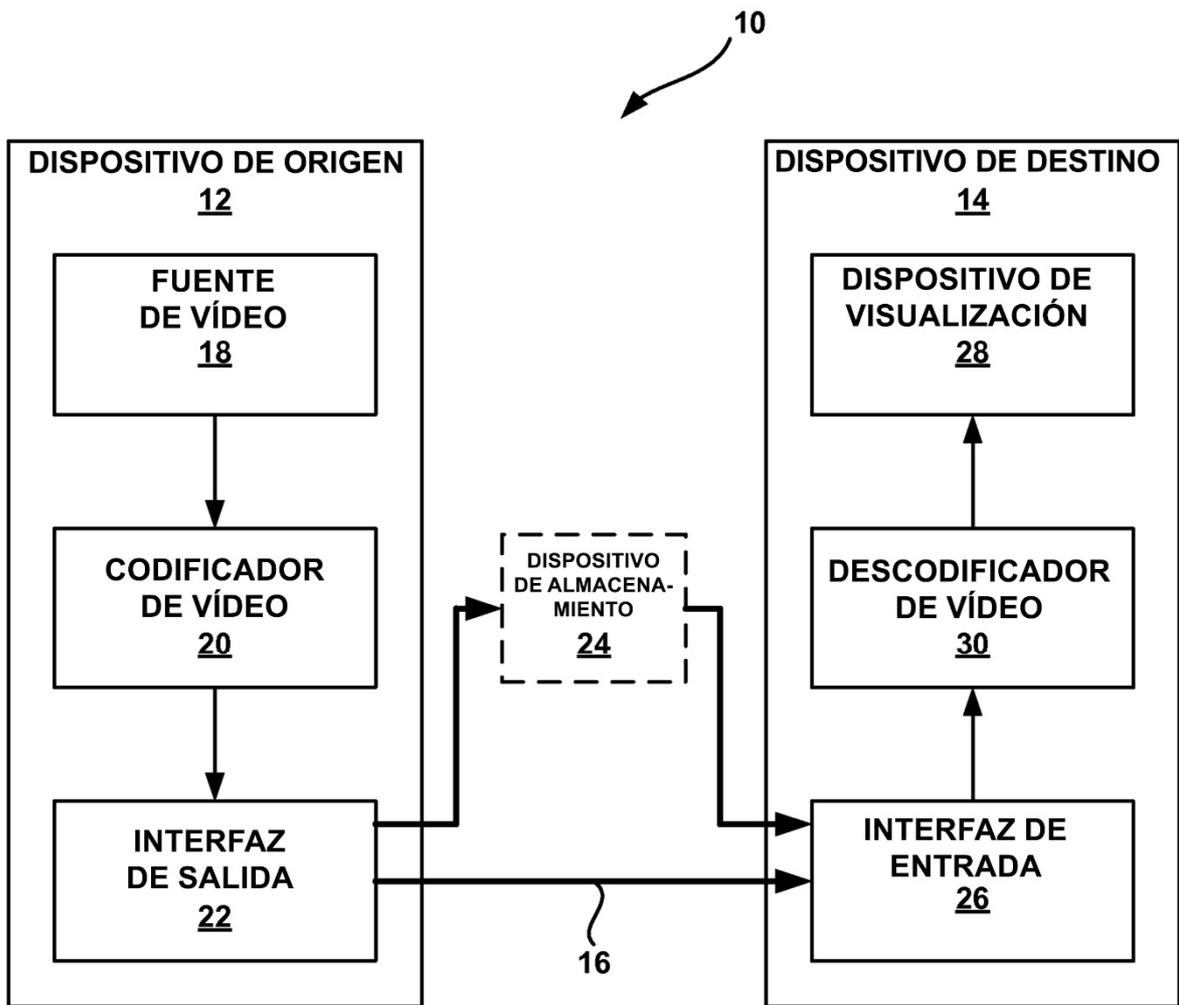


FIG. 1

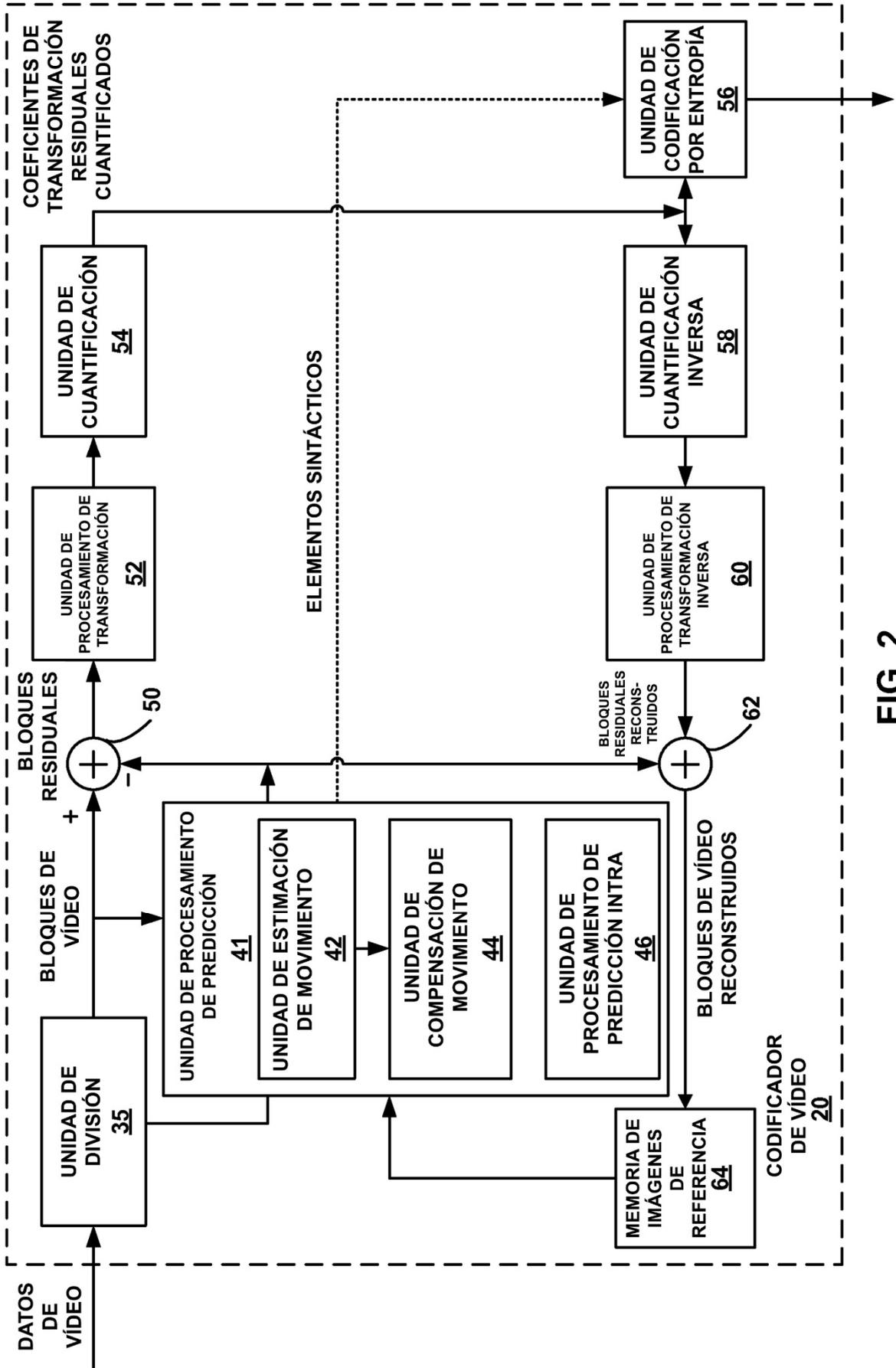


FIG. 2

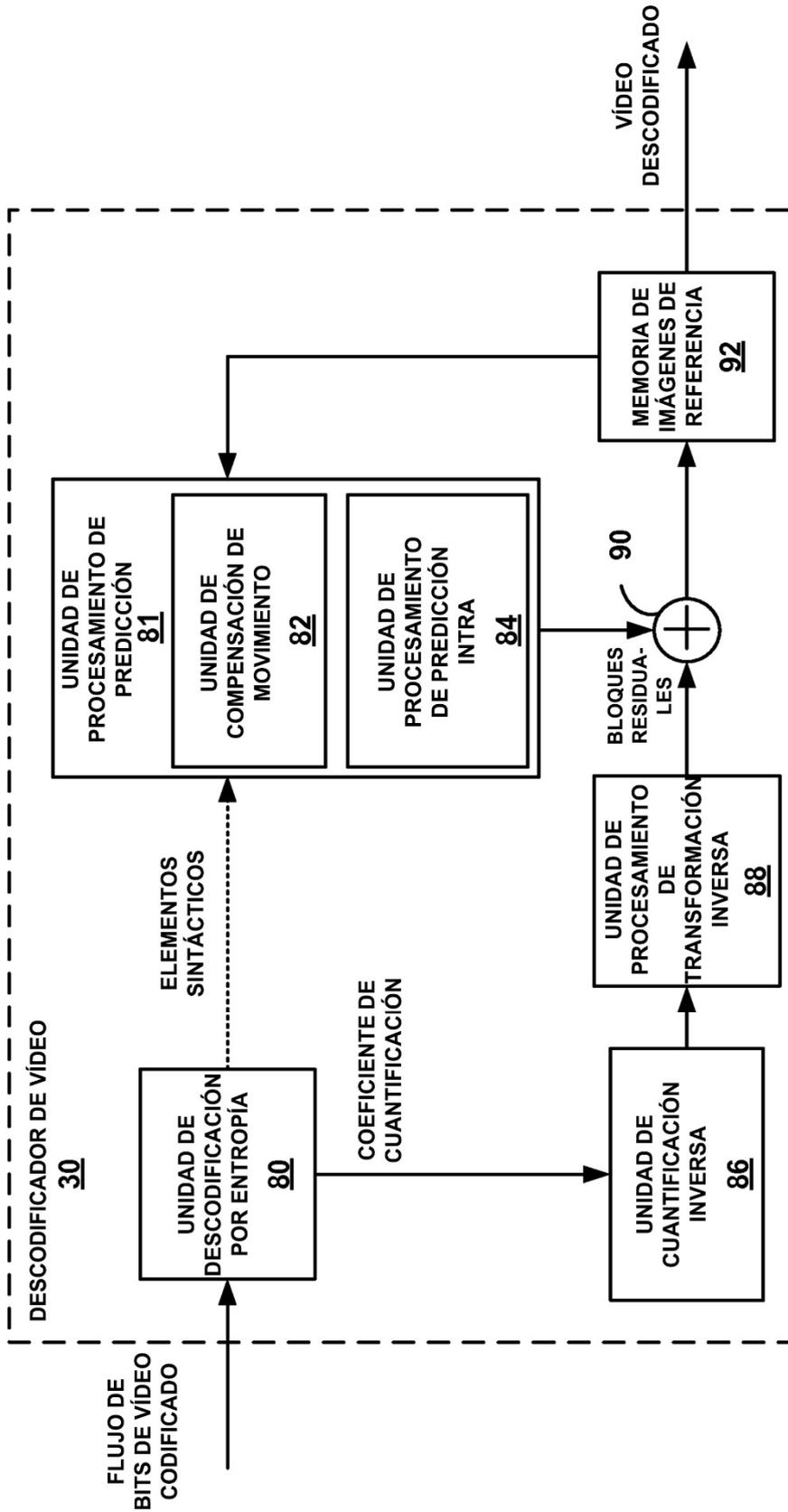


FIG. 3

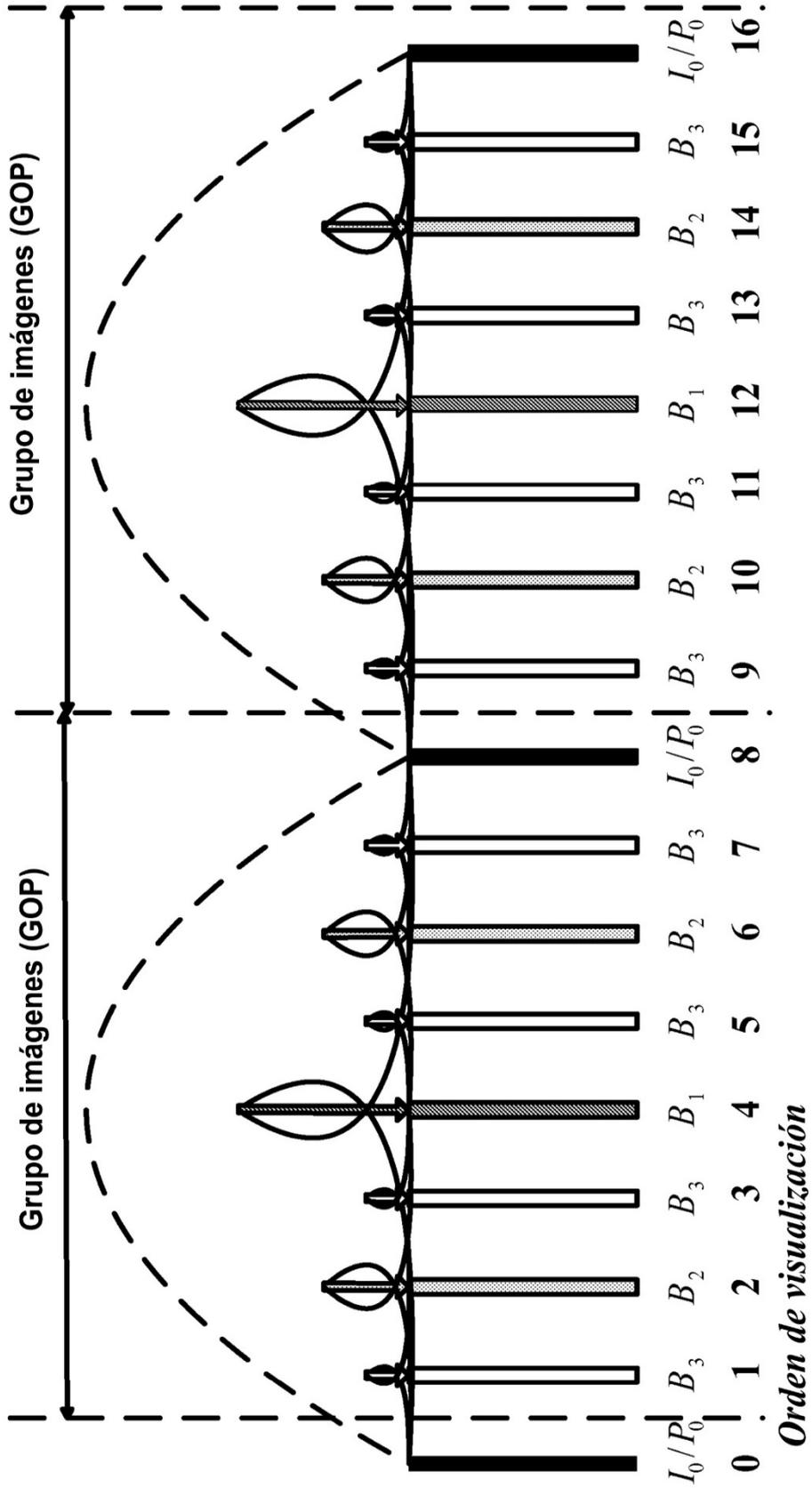


FIG. 4

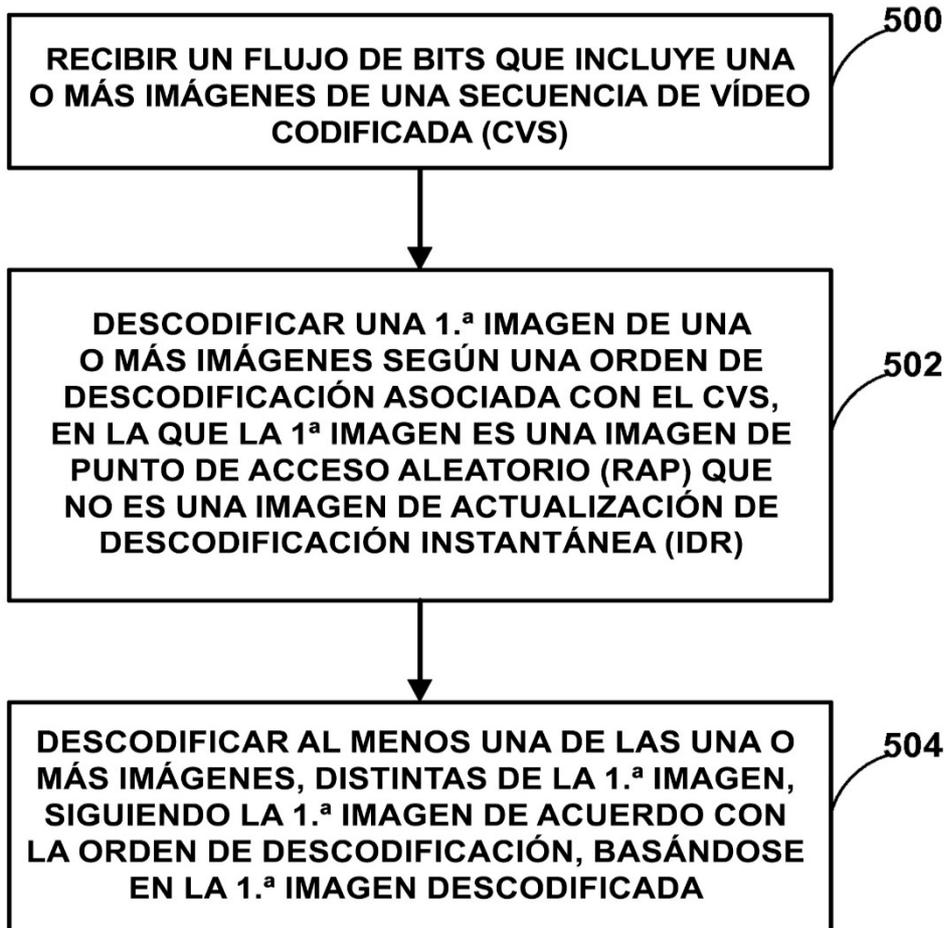


FIG. 5

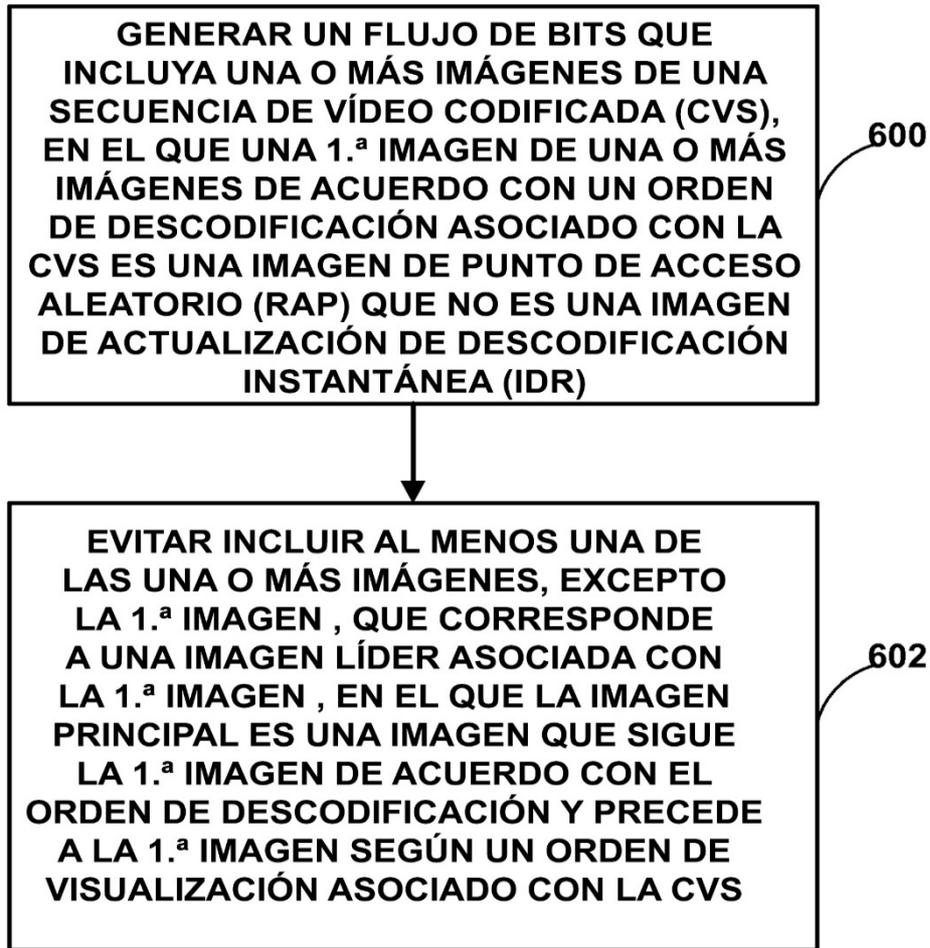


FIG. 6