

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 592 182**

51 Int. Cl.:

H04N 19/70 (2014.01)

H04N 19/46 (2014.01)

H04N 19/44 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **26.02.2013 PCT/US2013/027815**

87 Fecha y número de publicación internacional: **10.10.2013 WO13151635**

96 Fecha de presentación y número de la solicitud europea: **26.02.2013 E 13710663 (9)**

97 Fecha y número de publicación de la concesión europea: **15.06.2016 EP 2834975**

54 Título: **Almacenamiento intermedio de vídeo de bajo retardo en codificación de vídeo**

30 Prioridad:

04.04.2012 US 201261620266 P

01.05.2012 US 201261641063 P

25.02.2013 US 201313776140

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

28.11.2016

73 Titular/es:

QUALCOMM INCORPORATED (100.0%)

5775 Morehouse Drive

San Diego, CA 92121-1714, US

72 Inventor/es:

WANG, YE-KUI y

CHEN, YING

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 592 182 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

Almacenamiento intermedio de vídeo de bajo retardo en codificación de vídeo.

5 CAMPO TÉCNICO

Esta divulgación se refiere a la codificación de vídeo.

ANTECEDENTES

10 Las capacidades de vídeo digital pueden incorporarse en una amplia gama de dispositivos, incluyendo televisiones digitales, sistemas de radiodifusión directa digitales, sistemas de radiodifusión inalámbricos, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, cámaras digitales, dispositivos de grabación digitales, reproductores multimedia digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos celulares o de radio por satélite, dispositivos de videoconferencia, etc. Las normas de codificación de vídeo incluyen ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 o ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual e ITU-T H.264 (también conocida como ISO/IEC MPEG-4 AVC), incluyendo sus ampliaciones de codificación de vídeo escalable (SVC) y de codificación de vídeo multivista (MVC). Además, la codificación de vídeo de alta eficiencia (HEVC) es una norma de codificación de vídeo que está desarrollándose por el Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) del Grupo de Expertos en Codificación de Vídeo (VCEG) de ITU-T y el Grupo de Expertos en Imágenes en Movimiento (MPEG) de ISO/IEC. Además, un borrador reciente de la siguiente norma HEVC, denominado como "Borrador 6 de trabajo de la HEVC" o "WD6", se describe en el documento JCTVC-H1003, de Bross y col., titulado "High Efficiency Video Coding (HEVC) Text Specification Draft 6" ["Especificación textual de la Codificación de Vídeo de Alta Eficacia (HEVC), Borrador 6"], Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, octava conferencia: San José, California, Estados Unidos, febrero de 2012, que, a partir del martes, 1 de mayo de 2012 puede descargarse desde:

15
20
25
30 Las técnicas de compresión de vídeo llevan a cabo una predicción espacial y/o una predicción temporal para reducir o eliminar la redundancia presente en las secuencias de vídeo. Para la codificación de vídeo basada en bloques, una trama de vídeo o un fragmento pueden dividirse en macrobloques. Cada macrobloque se puede dividir aún más. Los macrobloques de una trama o fragmento intra-codificado (I) se codifican usando predicción espacial con respecto a macrobloques vecinos. Los macrobloques de una trama o fragmento inter-codificado (P o B) pueden usar predicción espacial con respecto a macrobloques vecinos de la misma trama o fragmento, o predicción temporal con respecto a otras tramas de referencia.

35
40 KAZUI K ET AL: "Market needs and practicality of sub-picture based CPB operation", 8. Conferencia de JCT; 99. Reunión MPEG; 1-2-2012 - 10-2-2012; SAN JOSE; (Joint Collaborative Team on Video Coding of ISO/IEC JTC1/SC29/WG11 and ITU-T SG.16), URL: <http://wftp3.itu.int/av-arch/jctvc-site/>, no.JCTVC-H0215, 21 de enero de 2012 (2012-01-21), XP030111242) divulga una operación basada en subimagen de la memoria intermedia de imagen codificada (CPB), con el tiempo de eliminación de una subimagen de reciente introducción en la especificación de HRD.

RESUMEN

45 En general, esta divulgación describe varias técnicas para lograr un retardo de códec reducido de forma interoperable. En un ejemplo, estas técnicas se pueden lograr a través de un comportamiento de memoria intermedia de imágenes codificadas (CPB) en base a subimagen genérica.

50 En un ejemplo, un procedimiento de codificación de datos de vídeo incluye almacenar una o más unidades de decodificación de datos de vídeo en una memoria intermedia de imagen codificada (CPB). El procedimiento incluye adicionalmente obtener un tiempo de eliminación de memoria intermedia respectivo para las una o más unidades de decodificación. El procedimiento incluye adicionalmente eliminar las unidades de decodificación de la CPB de acuerdo con el tiempo de eliminación de memoria intermedia obtenido para cada una de las unidades de decodificación. El procedimiento incluye adicionalmente determinar si la CPB funciona a nivel de unidad de acceso o nivel de subimagen. El procedimiento incluye adicionalmente codificar los datos de vídeo correspondientes a las unidades de decodificación eliminados. Si la CPB funciona a nivel de unidad de acceso, la codificación de los datos de vídeo comprende unidades de acceso comprendidas en las unidades de decodificación. Si la CPB funciona a nivel de subimagen, la codificación de los datos de vídeo comprende subconjuntos de unidades de acceso comprendidas en las unidades de decodificación.

60 En otro ejemplo, un dispositivo para codificar datos de vídeo está configurado para almacenar una o más unidades de decodificación de datos de vídeo en una memoria intermedia de imagen codificada (CPB). El dispositivo está configurado adicionalmente para obtener un tiempo de eliminación de memoria intermedia respectivo para las una o más unidades de decodificación. El dispositivo está configurado adicionalmente para eliminar las unidades de decodificación de la CPB de acuerdo con el tiempo de eliminación de memoria intermedia obtenido para cada una de las unidades de decodificación. El dispositivo está configurado adicionalmente para determinar si la CPB funciona

a nivel de unidad de acceso o nivel de subimagen. El dispositivo está configurado adicionalmente para codificar los datos de vídeo correspondientes a las unidades de decodificación eliminados. Si la CPB funciona a nivel de unidad de acceso, la codificación de los datos de vídeo comprende unidades de acceso comprendidas en las unidades de decodificación. Si la CPB funciona a nivel de subimagen, la codificación de los datos de vídeo comprende subconjuntos de unidades de acceso comprendidas en las unidades de decodificación.

En otro ejemplo, un aparato para codificar datos de datos de vídeo incluye medios para almacenar una o más unidades de decodificación de datos de vídeo en una memoria intermedia de imagen codificada (CPB). El aparato incluye adicionalmente medios para obtener un tiempo de eliminación de memoria intermedia respectivo para las una o más unidades de decodificación. El aparato incluye adicionalmente medios para eliminar las unidades de decodificación de la CPB de acuerdo con el tiempo de eliminación de memoria intermedia obtenido para cada una de las unidades de decodificación. El aparato incluye adicionalmente medios para determinar si la CPB funciona a nivel de unidad de acceso o nivel de subimagen. El aparato incluye adicionalmente medios para codificar los datos de vídeo correspondientes a las unidades de decodificación eliminados. Si la CPB funciona a nivel de unidad de acceso, la codificación de los datos de vídeo comprende unidades de acceso comprendidas en las unidades de decodificación. Si la CPB funciona a nivel de subimagen, la codificación de los datos de vídeo comprende subconjuntos de unidades de acceso comprendidas en las unidades de decodificación.

En otro ejemplo, un medio de almacenamiento legible por ordenador comprende instrucciones almacenadas en el mismo que, cuando se ejecutan, hacen que un procesador almacene una o más unidades de decodificación de datos de vídeo en una memoria intermedia de imagen codificada (CPB). Las instrucciones hacen adicionalmente que un procesador obtenga un tiempo de eliminación de memoria intermedia respectivo para las una o más unidades de decodificación. Las instrucciones hacen adicionalmente que un procesador elimine las unidades de decodificación de la CPB de acuerdo con el tiempo de eliminación de memoria intermedia obtenido para cada una de las unidades de decodificación. Las instrucciones hacen adicionalmente que un procesador determine si la CPB funciona a nivel de unidad de acceso o nivel de subimagen. Las instrucciones hacen adicionalmente que un procesador codifique los datos de vídeo correspondientes a las unidades de decodificación eliminadas. Si la CPB funciona a nivel de unidad de acceso, la codificación de los datos de vídeo comprende unidades de acceso comprendidas en las unidades de decodificación. Si la CPB funciona a nivel de subimagen, la codificación de los datos de vídeo comprende subconjuntos de unidades de acceso comprendidas en las unidades de decodificación.

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, los dibujos y las reivindicaciones.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

La figura 1 es un diagrama de bloques que ilustra un sistema de codificación y decodificación de vídeo ejemplar que puede utilizar técnicas para desbloquear bordes entre los bloques de vídeo, según técnicas de esta divulgación.

La figura 2 es un diagrama de bloques que ilustra un ejemplo de un codificador de vídeo que puede implementar técnicas para desbloquear bordes entre los bloques de vídeo, según técnicas de esta divulgación.

La figura 3 es un diagrama de bloques que ilustra un ejemplo de un decodificador de vídeo, que decodifica una secuencia de vídeo codificada, según técnicas de esta divulgación.

La figura 4 es un diagrama de bloques que ilustra un dispositivo de destino ejemplar que puede implementar cualquiera o todas las técnicas de esta divulgación.

La figura 5 es un diagrama de flujo que ilustra un procedimiento ejemplar que incluye la eliminación de unidades de decodificación de datos de vídeo de una memoria intermedia de imágenes de acuerdo con un tiempo de eliminación de memoria intermedia obtenido, según las técnicas de esta divulgación.

La figura 6 es un diagrama de flujo que ilustra otro procedimiento ejemplar que incluye la eliminación de unidades de decodificación de datos de vídeo de una memoria intermedia de imágenes de acuerdo con un tiempo de eliminación de memoria intermedia obtenido, según las técnicas de esta divulgación.

La figura 7 es un diagrama de flujo que ilustra otro procedimiento ejemplar de datos de vídeo de procesamiento que incluye la salida de una imagen recortada en un proceso de abultamiento, de acuerdo con técnicas de esta divulgación.

DESCRIPCIÓN DETALLADA

Las aplicaciones de vídeo pueden incluir la reproducción local, streaming, difusión/multidifusión y aplicaciones conversacionales. Las aplicaciones conversacionales pueden incluir videotelefonía y videoconferencia, y también se conocen como aplicaciones de bajo retardo. Las aplicaciones conversacionales requieren un retardo relativamente

bajo de extremo a extremo de la totalidad de los sistemas, es decir, el retardo entre el momento en que se captura una trama de vídeo y el momento en que se muestra la trama de vídeo. Típicamente, el retardo aceptable de extremo a extremo para aplicaciones conversacionales debe ser de menos de 400 milisegundos (ms), y se puede considerar muy bueno un retardo de extremo a extremo de alrededor de 150 ms. Cada etapa de procesamiento puede contribuir al retardo global de extremo a extremo, por ejemplo, retardo de captura, retardo de pre-procesamiento, retardo de codificación, retardo de transmisión, retardo de almacenamiento en memoria intermedia de recepción (para anti-vibración), retardo de decodificación, retardo de la salida de imagen decodificada, retardo post-procesamiento y retardo de visualización. Por lo tanto, típicamente, el retardo de códec (retardo de codificación, retardo de decodificación y retardo de salida de imagen decodificada) debe reducirse al mínimo en las aplicaciones conversacionales. En particular, la estructura de codificación debe garantizar que el orden de decodificación de las imágenes y el orden de salida son idénticos de tal forma que el retardo de salida de imagen decodificada es igual a cero.

Las normas de codificación de vídeo pueden incluir una especificación de un modelo de almacenamiento temporal de vídeo. En AVC y HEVC, el modelo de almacenamiento en memoria intermedia se refiere como un decodificador de referencia ficticio (HRD), que incluye un modelo de almacenamiento en memoria intermedia de tanto la memoria intermedia de imagen codificada (CPB) como la memoria intermedia de imagen decodificada (DPB), y los comportamientos de CPB y DPB se especifican matemáticamente. El HRD impone directamente restricciones sobre temporización, tamaños de almacén temporal y velocidades de bits diferentes, e impone indirectamente restricciones sobre características y estadísticas de flujos de bits. Un conjunto completo de parámetros de HRD incluye cinco parámetros básicos, el retardo inicial de eliminación de la CPB, el tamaño de la CPB, la velocidad de bits, el retardo inicial de salida del DPB y el tamaño de la DPB.

En la AVC y la HEVC, la conformidad del flujo de bits y la conformidad del decodificador se especifican como partes de la especificación del HRD. Aunque se nombra como un tipo de decodificador, el HRD se necesita típicamente en el sector codificador para garantizar la conformidad del flujo de bits, y habitualmente no se necesita en el sector decodificador. Se especifican dos tipos de conformidad de flujo de bits o de HRD, concretamente Tipo I y Tipo II. Además, se especifican dos tipos de conformidad de decodificador, la conformidad de decodificador de la temporización de salida y la conformidad de decodificador del orden de salida.

En los modelos de HRD de la AVC y la HEVC, la decodificación o la eliminación de la CPB está basada en unidades de acceso, y se supone que la decodificación de imágenes es instantánea. En aplicaciones prácticas, si un decodificador en conformidad sigue estrictamente los tiempos de decodificación señalizados, por ejemplo, en los mensajes de información de mejora complementaria (SEI) de temporización de imágenes, para iniciar la decodificación de unidades de acceso, entonces el momento más temprano posible para emitir una imagen decodificada específica es igual al momento de decodificación de esa imagen específica, más el tiempo necesario para decodificar esa imagen específica. A diferencia de los modelos AVC y HEVC HRD, el tiempo necesario para decodificar una imagen en el mundo real no es igual a cero. Los términos "instantánea" e "instantáneamente" como se usan a lo largo de toda esta divulgación pueden referirse a cualquier duración de tiempo que puede asumirse como instantánea en uno o más modelos de codificación o un aspecto idealizado de una cualquiera uno o más modelos de codificación, con el entendimiento de que esto puede variar de ser "instantánea" en un sentido físico o literal. Por ejemplo, para los fines de esta divulgación, una función o proceso pueden ser considerados como nominalmente "instantáneos" si esto tiene lugar en o dentro de un margen práctico de un tiempo lo más temprano posible hipotético o idealizado para la función o proceso a realizar. La sintaxis y nombres de las variables como se usan en el presente documento pueden entenderse en algunos ejemplos de acuerdo con su significado dentro del modelo HEVC.

Un comportamiento CPB basado en subimagen se propuso en "Enhancement on operation of coded picture buffer", Kazui et al., Joint Collaborative Team on Video Coding (JCT-VC) de ITU-T SG16 WP3 y ISO/IEC JTC1/SC29/WG11, 7ª reunión: Ginebra, CH 21-30, Nov. 2011, JCTVC-G188 (disponible en http://phenix.int-evry.fr/jct/doc_end_user/documents/7_Geneva/wg11/JCTVC-G188-v2.zip) con el fin de lograr un retardo de codificación de menos de un periodo de imagen en una forma interoperable. El procedimiento JCTVC-G188 se puede resumir como se indica a continuación: una imagen puede dividirse uniformemente M grupos de bloques de árbol, es decir, los primeros M bloques de árbol en una exploración de trama de bloque de árbol de la imagen pertenecen al primer grupo de bloques de árbol, los segundos M bloques de árbol en la exploración de trama de bloque de árbol de la imagen pertenecen al segundo grupo de bloques de árbol, y así sucesivamente. El valor M se puede señalar en los mensajes SEI del periodo de almacenamiento intermedio. Este valor puede ser usado para derivar el tiempo de eliminación de CPB (es decir, el tiempo de decodificación) de cada grupo de bloques de árbol. En este sentido, el comportamiento JCTVC-G188 CPB se basa en subimagen, donde cada subimagen es un grupo de bloques de árbol. En algunos ejemplos, una subimagen puede corresponder a uno o más fragmentos, uno o más ondas (para la partición del frente de onda de una imagen), o una o más piezas. Se asume en este procedimiento de JCTVC-G188 que los tiempos de eliminación de CPB a nivel de unidad de acceso se señalizan como habituales (usando mensajes SEI de temporización de imagen), y dentro de cada unidad de acceso, los tiempos de eliminación de CPB para los grupos bloques de árbol se asume que dividen de forma lineal o uniforme el intervalo del tiempo de eliminación de CPB de la unidad de acceso anterior al tiempo de eliminación de CPB de la unidad de acceso actual.

Este procedimiento de JCTVC-G188 implica además las siguientes suposiciones o requisitos de flujo de bits: (1) dentro de cada imagen, cada grupo de bloques de árbol se codifica de una manera que requiere la misma cantidad de tiempo de decodificación (no sólo en el modelo HRD, sino también para los decodificadores del mundo real), donde los datos codificados del primer grupo de bloques de árbol se considera que incluyen todas las unidades NAL (capa de abstracción de red) no VCL (capa de codificación de vídeo) en la misma unidad de acceso y antes de la primera unidad NAL VCL; (2) dentro de cada imagen, el número de bits para cada grupo de bloques de árbol es idéntico, en el que los datos codificados del primer grupo de bloques de árbol se considera que incluyen todas las unidades NAL no VCL en la misma unidad de acceso y antes de la primera unidad NAL VCL.

Los procedimientos existentes para la especificación de un comportamiento CPB basado en subimagen están asociados a al menos los siguientes problemas: (1) El requisito de que la cantidad de datos codificados para cada grupo de bloques de árbol en una imagen codificada es idéntica y difícil de lograr con un rendimiento de codificación equilibrado (donde los grupos de bloques de árbol para áreas con una textura más detallada o actividad de movimiento en una imagen pueden usar más bits). (2) Cuando se incluye más de un grupo de bloques de árbol en una porción, no puede haber ninguna manera fácil de dividir los bits codificados de bloques de árbol pertenecientes a diferentes grupos de bloques de árbol y enviarlos por separado en el lado del codificador y eliminarlos por separado de la CPB (es decir, decodificarlos por separado).

Para hacer frente a los problemas anteriores, esta divulgación describe un diseño genérico para el apoyo del comportamiento CPB basado en subimagen, con diversas alternativas. En algunos ejemplos, las características de las técnicas CPB basadas en subimagen de esta divulgación pueden incluir aspectos de las siguientes técnicas: (1) Cada subimagen puede incluir varios bloques de una imagen codificada continua en orden de decodificación. Un bloque de codificación puede ser idéntico a un bloque de árbol o un subconjunto de un bloque de árbol; (2) La codificación de subimágenes y la asignación de bits en diferentes subimágenes en una imagen puede realizarse como de costumbre, sin asumir o requerir que cada subimagen (es decir, grupo de bloques de árbol) en una imagen se codifica con la misma cantidad de bits. En consecuencia, el tiempo de eliminación de la CPB para cada subimagen se puede señalar en el flujo de bits en lugar de derivarse de acuerdo a los tiempos de eliminación de CPB a nivel de imagen señalados; (3) Cuando se incluye más de una subimagen en una porción, la alineación de bytes se puede aplicar al final de cada subimagen, por el contrario, por ejemplo, de la alineación de bytes para las piezas en HEVC WD6. Además, el punto de entrada de cada subimagen, excepto para el primero en la imagen codificada, se puede señalar, a diferencia, de la alineación de byte para las piezas en HEVC WD6. El valor señalado recibido puede ser indicativo de la alineación de bytes de al menos una de las subimágenes dentro de un conjunto mayor de los datos de vídeo, tales como una porción, una pieza, o una trama, por ejemplo. Cada una de las características (1)-(3) se puede aplicar de forma independiente o en combinación con las demás.

En un ejemplo, la operación HRD, incluyendo el comportamiento CPB basado en subimagen, se puede resumir como se indica a continuación: Cuando la señalización indica que el comportamiento CPB basado en subimagen está en uso, por ejemplo, a través de una señalización a nivel de secuencia de un elemento de sintaxis `sub_pic_cpb_flag` igual a 1, la eliminación o decodificación de la CPB se basa en la subimagen, o de forma equivalente, la unidad de decodificación, que puede ser una unidad de acceso o un subconjunto de una unidad de acceso. En otras palabras, cada vez que una unidad de decodificación, ya sea una unidad de acceso o un subconjunto de una unidad de acceso, se elimina de la CPB para la decodificación, el tiempo de eliminación de una unidad de decodificación de la CPB puede obtenerse a partir de un retardo de eliminación de CPB inicial señalado y el retardo de la eliminación de CPB señalado para la unidad de decodificación. Se especifica una CPB de flujo inferior como la condición en la que el tiempo de eliminación de CPB nominal de la unidad de decodificación m $t_{r,n}(m)$ es menor que el tiempo de eliminación de CPB final de la unidad de decodificación m $t_{af}(m)$ para cualquier valor de m . En un ejemplo, cuando un elemento de sintaxis `lowdelayhrdflag` es igual a 0, se requiere que la CPB no esté nunca con un flujo inferior.

En un ejemplo, los procesos de salida y eliminación de DPB todavía pueden operar a nivel de imagen o nivel de unidad de acceso, es decir, cada vez que toda la imagen decodificada se emite o se elimina de la DPB. La eliminación de imágenes decodificadas de la DPB puede suceder instantáneamente en el tiempo de eliminación de la CPB de la primera unidad de decodificación de la unidad de acceso n (que contiene la imagen actual).

La figura 1 es un diagrama de bloques que ilustra un sistema de codificación y decodificación de vídeo ejemplar 10 que puede utilizar técnicas para el almacenamiento de una o más unidades de decodificación de datos de vídeo en una memoria intermedia de imágenes, para obtener un tiempo de eliminación de memoria intermedia respectivo para las una o más unidades de decodificación, eliminar las unidades de decodificación de la memoria intermedia de imágenes de acuerdo con el tiempo de eliminación de memoria intermedia obtenido para cada una de las unidades de decodificación, y codificar datos de vídeo correspondientes a las unidades de decodificación eliminadas, entre otras funciones.

Como se muestra en la figura 1, el sistema 10 incluye un dispositivo de origen 12 que transmite vídeo codificado a un dispositivo de destino 14 a través de un canal de comunicación 16. El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender cualquiera de una amplia gama de dispositivos. En algunos casos, el dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender dispositivos de comunicación inalámbricos, tales como

microteléfonos inalámbricos, los denominados radioteléfonos celulares o vía satélite, o cualquier dispositivo inalámbrico que pueda comunicar información de vídeo a través de un canal de comunicación 16, donde el canal de comunicación 16 es inalámbrico en este caso. Sin embargo, las técnicas de esta divulgación no están limitadas necesariamente a aplicaciones o configuraciones inalámbricas. Por ejemplo, estas técnicas pueden aplicarse a las radiodifusiones televisivas inalámbricas, a las transmisiones televisivas por cable, a las transmisiones televisivas vía satélite, a las transmisiones de vídeo por Internet, al vídeo digital codificado que está codificado en un medio de almacenamiento y a otros escenarios. Por consiguiente, el canal de comunicación 16 puede comprender cualquier combinación de medios inalámbricos, cableados o de almacenamiento adecuados para la transmisión o el almacenamiento de datos de vídeo codificados.

Como alternativa, los datos codificados pueden transmitirse del transmisor 24 a un dispositivo de almacenamiento 34. De forma similar, a los datos codificados se puede acceder desde el dispositivo de almacenamiento 34 mediante el receptor 26. El dispositivo de almacenamiento 34 puede incluir cualquiera de una diversidad de medios de almacenamiento de datos de acceso distribuido o local, tales como una unidad de disco, discos Blu-ray, discos DVD, discos CD-ROM, memoria flash, memoria volátil o no volátil, u otros medios adecuados cualesquiera de almacenamiento digital, para almacenar datos de vídeo codificados. En un ejemplo adicional, el dispositivo de almacenamiento 34 puede corresponder a un servidor de ficheros, un servidor virtual, un centro de datos, una red redundante de centros de datos, u otro dispositivo de almacenamiento intermedio que pueda retener el vídeo codificado generado por el dispositivo de origen 12. El dispositivo de destino 14 puede acceder a datos de vídeo almacenados procedentes del dispositivo de almacenamiento 34, mediante flujos de transmisión o descarga. Una implementación del servidor de ficheros del dispositivo de almacenamiento 34, o una porción del mismo, puede ser cualquier tipo de servidor capaz de almacenar datos de vídeo codificados y transmitir esos datos de vídeo codificados al dispositivo de destino 14. Los servidores ejemplares de ficheros incluyen un servidor de la Red (por ejemplo, para una sede de la Red), un servidor del FTP, dispositivos de almacenamiento conectados a la red (NAS) o una unidad de disco local. El dispositivo de destino 14 puede acceder a los datos de vídeo codificados a través de cualquier conexión de datos estándar, incluyendo una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión de Wi-Fi), una conexión cableada (por ejemplo, DSL, módem de cable, etc.), o una combinación de ambos que sea adecuada para acceder a datos de vídeo codificado almacenados en un dispositivo de almacenamiento remoto o no local 34. La transmisión de datos de vídeo codificado desde el dispositivo de almacenamiento 34 puede ser una transmisión por flujo, una transmisión de descarga o una combinación de ambas.

En el ejemplo de la figura 1, el dispositivo de origen 12 incluye una fuente de vídeo 18, un codificador de vídeo 20, un modulador/demodulador (módem) 22 y un transmisor 24. El dispositivo de destino 14 incluye un receptor 26, un módem 28, un decodificador de vídeo 30 y un dispositivo de visualización 32. De acuerdo con esta divulgación, el codificador de vídeo 20 del dispositivo de origen 12 puede configurarse para aplicar las técnicas para el almacenamiento de una o más unidades de decodificación de datos de vídeo en una memoria intermedia de imágenes, para obtener un tiempo de eliminación de memoria intermedia respectivo para las una o más unidades de decodificación, eliminar las unidades de decodificación de la memoria intermedia de imágenes de acuerdo con el tiempo de eliminación de memoria intermedia obtenido para cada una de las unidades de decodificación, y codificar datos de vídeo correspondientes a las unidades de decodificación eliminadas, entre otras funciones. En otros ejemplos, un dispositivo de origen y un dispositivo de destino pueden incluir otros componentes o disposiciones. Por ejemplo, el dispositivo de origen 12 puede recibir datos de vídeo de una fuente de vídeo externa en lugar de una fuente de vídeo integrada 18, tal como una cámara externa. Asimismo, el dispositivo de destino 14 puede interactuar con un dispositivo de visualización externo, en lugar de incluir un dispositivo de visualización integrado 32.

El sistema ilustrado 10 de la figura 1 es simplemente un ejemplo. Las técnicas para el almacenamiento de una o más unidades de decodificación de datos de vídeo en una memoria intermedia de imágenes, para obtener un tiempo de eliminación de memoria intermedia respectivo para las una o más unidades de decodificación, eliminar las unidades de decodificación de la memoria intermedia de imágenes de acuerdo con el tiempo de eliminación de memoria intermedia obtenido para cada una de las unidades de decodificación, y codificar datos de vídeo correspondientes a las unidades de decodificación eliminadas pueden realizarse por cualquier dispositivo de codificación y/o decodificación de vídeo digital. Aunque, por lo general, las técnicas de esta divulgación se llevan a cabo por un dispositivo de codificación de vídeo, las técnicas también pueden llevarse a cabo mediante un codificador/decodificador de vídeo, denominado habitualmente "códec". Además, las técnicas de esta divulgación también pueden llevarse a cabo por un preprocesador de vídeo. El dispositivo de origen 12 y el dispositivo de destino 14 son simplemente ejemplos de tales dispositivos de codificación, donde el dispositivo de origen 12 genera datos de vídeo codificados para su transmisión al dispositivo de destino 14. En algunos ejemplos, los dispositivos 12, 14 pueden funcionar de manera casi simétrica, de modo que cada uno de los dispositivos 12, 14 incluye componentes de codificación y de decodificación de vídeo. Por tanto, el sistema 10 puede soportar una transmisión de vídeo unidireccional o bidireccional entre los dispositivos de vídeo 12, 14, por ejemplo para el flujo continuo de vídeo, la reproducción de vídeo, la radiodifusión de vídeo o la videotelefonía.

La fuente de vídeo 18 del dispositivo de origen 12 puede incluir un dispositivo de captura de vídeo, tal como una cámara de vídeo, un archivo de vídeo que contiene vídeo grabado previamente y/o vídeo procedente de un proveedor de contenido de vídeo. Como una alternativa adicional, la fuente de vídeo 18 puede generar vídeo fuente como datos basados en gráficos de ordenador, o una combinación de vídeo en directo, vídeo de un archivo y vídeo

generado por ordenador. En algunos casos, 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, como se ha mencionado anteriormente, las técnicas descritas en esta divulgación pueden aplicarse a la codificación de vídeo en general, y pueden aplicarse a aplicaciones inalámbricas y/o cableadas. En cada caso, el vídeo grabado, pregrabado o generado por ordenador puede codificarse por el codificador de vídeo 20. La información de vídeo codificado puede modularse después por el módem 22, según una norma de comunicación, y transmitirse al dispositivo de destino 14 a través del transmisor 24. El módem 22 puede incluir varios mezcladores, filtros, amplificadores u otros componentes diseñados para la modulación de señales. El transmisor 24 puede incluir circuitos diseñados para transmitir datos, incluyendo amplificadores, filtros y una o más antenas.

El receptor 26 del dispositivo de destino 14 recibe información a través del canal 16, y el módem 28 desmodula la información. De nuevo, el proceso de codificación de vídeo puede implementar una o más de las técnicas descritas en el presente documento para almacenar una o más unidades de decodificación de datos de vídeo en una memoria intermedia de imágenes, para obtener un tiempo de eliminación de memoria intermedia respectivo para las una o más unidades de decodificación, eliminar las unidades de decodificación de la memoria intermedia de imágenes de acuerdo con el tiempo de eliminación de memoria intermedia obtenido para cada una de las unidades de decodificación, y codificar datos de vídeo correspondientes a las unidades de decodificación eliminadas, entre otras funciones. La información comunicada a través del canal 16 puede incluir información sintáctica definida por el codificador de vídeo 20, que también puede usarse por el decodificador de vídeo 30, que incluye elementos sintácticos que describen características y/o el procesamiento de macrobloques, unidades de árbol de codificación, porciones, u otras unidades codificadas, por ejemplo grupos de imágenes (GOP). El dispositivo de visualización 32 muestra los datos de vídeo decodificados a un usuario y puede comprender cualquiera de una variedad de dispositivos de visualización, tal como un tubo de rayos catódicos (CRT), una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodos orgánicos de emisión de luz (OLED) u otro tipo de dispositivo de visualización.

En el ejemplo de la figura 1, el canal de comunicación 16 puede comprender cualquier medio de comunicación inalámbrico o cableado, tal como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión física, o cualquier combinación de medios inalámbricos y cableados. El canal de comunicación 16 puede formar parte de una red basada en paquetes, tal como una red de área local, una red de área extensa o una red global tal como Internet. El canal de comunicación 16 representa generalmente cualquier medio de comunicación adecuado, o un conjunto de diferentes medios de comunicación, para transmitir datos de vídeo desde el dispositivo de origen 12 hasta el dispositivo de destino 14, incluyendo cualquier combinación adecuada de medios cableados o inalámbricos. El canal de comunicación 16 puede incluir encaminadores, conmutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo de origen 12 al dispositivo de destino 14. En otros ejemplos, el dispositivo de origen 12 puede almacenar datos codificados en un medio de almacenamiento, tal como en el dispositivo de almacenamiento 34, en lugar de transmitir los datos. Asimismo, el dispositivo de destino 14 puede estar configurado para recuperar datos codificados del dispositivo de almacenamiento 34 o un medio o dispositivo de almacenamiento.

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar según una norma de compresión de vídeo, tal como las descritas en el presente documento. Sin embargo, las técnicas de esta divulgación no están limitadas a ninguna norma de codificación particular. Aunque no se muestra en la figura 1, en algunos aspectos, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden estar integrados en un codificador y un decodificador de audio, y pueden incluir unidades de multiplexación y desmultiplexación adecuadas, u otro hardware y software, para llevar a cabo la codificación tanto de audio como de vídeo en un flujo de datos común o en flujos de datos diferentes. Si procede, las unidades de multiplexación y desmultiplexación pueden ajustarse al protocolo de multiplexación ITU H.223 o a otros protocolos, tales como el protocolo de datagramas de usuario (UDP).

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden implementarse como cualquiera de una variedad de sistemas de circuitos de codificación adecuados, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados de aplicación específica (ASIC), matrices de puertas de campo programable (FPGA), dispositivos de comunicación inalámbricos que incluyen un dispositivo de codificación de vídeo, tal como un codificador o un decodificador, lógica discreta, software, hardware, firmware o cualquier combinación de los mismos. El codificador de vídeo 20 y el decodificador de vídeo 30 pueden incluirse en uno o más codificadores o decodificadores, donde cualquiera de los mismos puede estar integrado como parte de un codificador/decodificador combinado (códec) en una cámara respectiva, ordenador, dispositivo móvil, dispositivo de abonado, dispositivo de radiodifusión, codificador, servidor, u otro dispositivo.

Una secuencia de vídeo incluye normalmente una serie de tramas de vídeo. Un grupo de imágenes (GOP) comprende generalmente una serie de una o más tramas de vídeo. Un GOP puede incluir datos sintácticos en la cabecera del GOP, en la cabecera de uno o más cuadros del GOP, o en otras ubicaciones, que indican el número de tramas incluidas en el GOP. Cada trama puede incluir datos sintácticos de trama que describen el modo de codificación de la trama respectiva. Un codificador de vídeo 20 actúa habitualmente sobre bloques de vídeo, también denominados como unidades de codificación (CU), en las tramas de vídeo individuales con el fin de codificar los datos de vídeo. Un bloque de vídeo puede corresponder a una unidad de codificación mayor (LCU) o una partición

de una LCU. Los bloques de vídeo pueden tener tamaños fijos o variables y pueden tener un tamaño diferente según una norma de codificación específica. Cada trama de vídeo puede incluir una pluralidad de fragmentos. Cada fragmento puede incluir una pluralidad de LCU, que pueden disponerse en particiones, también denominadas como subCUs. Una LCU también puede denominarse como una unidad de árbol de codificación.

5 Como un ejemplo, la norma ITU-T H.264 soporta intra-predicción para varios tamaños de bloque, tales como 16 x 16, 8 x 8 o 4 x 4 para componentes de luminancia, y 8 x 8 para componentes de crominancia, así como inter-
 10 predicción en varios tamaños de bloque, tales como 16 x 16, 16 x 8, 8 x 16, 8 x 8, 8 x 4, 4 x 8 y 4 x 4 para componentes de luminancia y tamaños escalados correspondientes para componentes de crominancia. En esta divulgación, "NxN" y "N por N" pueden usarse de manera intercambiable para hacer referencia a las dimensiones de
 15 píxel del bloque en lo que respecta a la dimensión vertical y la dimensión horizontal, por ejemplo 16x16 píxeles o 16 por 16 píxeles. En general, un bloque 16 x 16 tendrá 16 píxeles en la dirección vertical ($y = 16$) y 16 píxeles en la dirección horizontal ($x = 16$). Asimismo, un bloque NxN tiene generalmente N píxeles en la dirección vertical y N píxeles en la dirección horizontal, donde N representa un valor entero no negativo. Los píxeles de un bloque pueden estar ordenados en filas y columnas. Además, los bloques no necesitan tener necesariamente el mismo número de píxeles en la dirección horizontal y en la dirección vertical. Por ejemplo, los bloques pueden comprender N x M píxeles, donde M no es necesariamente igual a N.

20 Los bloques de vídeo pueden comprender bloques de datos de píxel en el dominio de píxel, o bloques de coeficientes de transformada en el dominio de transformada, por ejemplo, tras la aplicación de una transformada, tal como una transformada discreta del coseno (DCT), una transformada de número entero, una transformada de
 25 ondícula o una transformada conceptualmente similar, a los datos residuales de bloque de vídeo que representan diferencias de píxel entre bloques de vídeo codificados y bloques de vídeo predictivos. En algunos casos, un bloque de vídeo puede comprender bloques de coeficientes de transformada cuantificados en el dominio de transformada.

30 Bloques de vídeo más pequeños pueden proporcionar una mejor resolución y pueden usarse en ubicaciones de una trama de vídeo que incluyen altos niveles de detalle. En general, los bloques y las diversas particiones, denominadas en ocasiones subbloques, pueden considerarse bloques de vídeo. Además, un fragmento puede considerarse una pluralidad de bloques de vídeo, tales como bloques y/o subbloques. Cada fragmento puede ser
 35 una unidad de una trama de vídeo que puede decodificarse de manera independiente. Como alternativa, las propias tramas pueden ser unidades decodificables, u otras partes de un cuadro pueden definirse como unidades decodificables. El término "unidad codificada" puede referirse a cualquier unidad de una trama de vídeo que puede decodificarse de manera independiente, tal como una trama completa o un fragmento de una trama, a un grupo de imágenes (GOP) denominado también como una secuencia de vídeo codificada, o a otra unidad que pueda decodificarse de manera independiente definida según técnicas de codificación aplicables.

40 Tras una codificación intra-predictiva o inter-predictiva para generar datos predictivos y datos residuales, y tras cualquier transformada (tal como la transformada de número entero 4x4 u 8x8 usada en la norma H.264/AVC o una transformada discreta del coseno DCT) para generar coeficientes de transformada, puede llevarse a cabo la cuantificación de los coeficientes de transformada. La cuantificación se refiere generalmente a un proceso en el que los coeficientes de transformada se cuantifican para reducir posiblemente la cantidad de datos usados para representar los coeficientes. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos o todos los coeficientes. Por ejemplo, un valor de n bits puede redondearse por lo bajo a un valor de m bits durante la cuantificación, donde n es mayor que m .

45 HECV se refiere a un bloque de datos de vídeo como una unidad de codificación, que puede incluir una o más unidades de predicción (PU) y/o una o más unidades de transformada (TU). Esta divulgación también puede usar el término "bloque" para referirse a una CU, una PU o una TU. Los datos sintácticos de un flujo de bits pueden definir la unidad de codificación más grande (LCU), que es la mayor unidad de codificación en lo que respecta al número de
 50 píxeles. En general, una CU tiene un propósito similar a un macrobloque de H.264, excepto que una CU no tiene una distinción tamaño. Por lo tanto, una CU puede dividirse en subCUs. En general, las referencias de esta divulgación a una CU pueden referirse a la unidad de codificación más grande de una imagen o a una subCU de una LCU. Una LCU puede dividirse en varias subCUs y cada subCU puede dividirse adicionalmente en varias subCUs. Los datos sintácticos de un flujo de bits pueden definir el número máximo de veces que puede dividirse una LCU, lo que se denomina profundidad de CU. Por consiguiente, los datos sintácticos también pueden definir la unidad de codificación menor (SCU).

55 Una LCU puede asociarse a una estructura de datos de árbol cuaternario. En general, una estructura de datos de árbol cuaternario incluye un nodo por CU, donde el nodo raíz corresponde a la LCU. Si una CU se divide en cuatro subCUs, el nodo correspondiente a la CU incluye cuatro nodos hoja, cada uno de los cuales corresponde a una de las subCUs. Cada nodo de la estructura de datos de árbol cuaternario puede proporcionar datos sintácticos para la CU correspondiente. Por ejemplo, un nodo del árbol cuaternario puede incluir un indicador de división, que indica si la CU correspondiente al nodo está dividida en varias subCUs. Los elementos sintácticos de una CU pueden definirse de manera recursiva y pueden depender de si la CU está dividida en varias subCUs. Si una CU no está dividida adicionalmente, se denomina CU hoja. En esta divulgación, 4 subCUs de una CU hoja también se denominarán CU hoja aunque no haya una división explícita de la CU hoja original. Por ejemplo, si una CU con un
 60
 65

tamaño de 16 x 16 no se divide adicionalmente, las cuatro subCUs 8 x 8 también se denominarán CU hoja aunque la CU 16 x 16 no se haya dividido.

Además, las TU de las CUs hoja también pueden asociarse a estructuras de datos respectivas de árbol cuaternario. Es decir, una CU hoja puede incluir un árbol cuaternario que indica cómo la CU hoja está dividida en varias TU. Esta divulgación hace referencia al árbol cuaternario que indica cómo una LCU está dividida como un árbol cuaternario de CU, indicando el árbol cuaternario cómo una CU hoja está dividida en varias TU como un árbol cuaternario de TU. El nodo raíz de un árbol cuaternario de TU corresponde generalmente a una CU hoja, mientras que el nodo raíz de un árbol cuaternario de CU corresponde generalmente a una LCU. Las TUs del árbol cuaternario de TU que no están divididas pueden denominarse TU hoja.

Una CU hoja puede incluir una o más unidades de predicción (PUs). En general, una PU representa la totalidad o una parte de la CU correspondiente, y puede incluir datos para recuperar una muestra de referencia para la PU. Por ejemplo, cuando la PU está codificada en el modo de inter-predicción, la PU puede incluir datos que definen un vector de movimiento para la PU. Los datos que definen el vector de movimiento pueden describir, por ejemplo, un componente horizontal del vector de movimiento, un componente vertical del vector de movimiento, una resolución para el vector de movimiento (por ejemplo, precisión de píxeles de un cuarto o precisión de píxeles de un octavo), una trama de referencia a la que apunta el vector de movimiento y/o una lista de referencia (por ejemplo, la Lista 0 o la Lista 1) para el vector de movimiento. Los datos de la CU hoja que definen la(s) PU(s) también pueden describir, por ejemplo, una división de la CU en una o más PUs. Los modos de división pueden diferir en función de si la CU está sin codificar, del modo de intra-predicción codificado o del modo de inter-predicción codificado. En la intra-codificación, una PU puede tratarse de mismo modo que una unidad de transformada de hoja descrita a continuación.

Una CU hoja puede incluir una o más unidades de transformada (TUs). Las unidades de transformada pueden especificarse usando una estructura de árbol cuaternario de TU, como la descrita anteriormente. Es decir, un indicador de división puede indicar si una CU hoja está dividida en cuatro unidades de transformada. Después, cada unidad de transformada puede dividirse adicionalmente en 4 subTUs. Cuando una TU no está dividida adicionalmente, puede denominarse como TU hoja. En general, un indicador de división puede indicar que una TU hoja está dividida en varias TUs de forma cuadrada. Con el fin de indicar que una TU está dividida en varias TUs que no tienen una forma cuadrada, pueden incluirse otros datos sintácticos, por ejemplo datos sintácticos que indican que las TUs van a dividirse según la transformada de árbol cuaternario no cuadrada (NSQT).

Generalmente, en lo que respecta a la intra-codificación, todas las TU hoja que pertenecen a una CU hoja comparten el mismo modo de intra-predicción. Es decir, el mismo modo de intra-predicción se aplica generalmente para calcular valores predichos para todas las TUs de una CU hoja. En lo que respecta a la intra-codificación, un codificador de vídeo puede calcular un valor residual para cada TU hoja usando el modo de intra-predicción, como una diferencia entre la parte de los valores predictivos correspondientes a la TU y el bloque original. El valor residual puede transformarse, cuantificarse y examinarse. En lo que respecta a la inter-codificación, un codificador de vídeo puede llevar a cabo la predicción en el nivel de PU y puede calcular un valor residual para cada PU. Los valores residuales correspondientes a una CU hoja pueden transformarse, cuantificarse y examinarse. En lo que respecta a la inter-codificación, una TU hoja puede ser mayor o menor que una PU. En lo que respecta a la intra-codificación, una PU puede estar cubricada con una TU hoja correspondiente. En algunos ejemplos, el tamaño máximo de una TU hoja puede ser el tamaño de la CU hoja correspondiente.

En general, esta divulgación usa los términos CU y TU para hacer referencia a una CU hoja y a una TU hoja, respectivamente, a no ser que se indique lo contrario. En general, las técnicas de esta divulgación se refieren a transformar, cuantificar, escanear y codificar por entropía datos de una CU. A modo de ejemplo, las técnicas de esta divulgación incluyen la selección de una transformada a utilizar para transformar un valor residual de un bloque de intra-predicción basado en un modo de intra-predicción utilizado para predecir el bloque. Esta divulgación también usa la expresión "transformada direccional" o "transformada diseñada" para referirse a tal transformada que depende de la dirección del modo intra-predicción. Es decir, un codificador de vídeo puede seleccionar una transformada direccional para aplicarse a una unidad de transformada (TU). Como se ha señalado anteriormente, la intra-predicción incluye la predicción de la TU de una CU actual de una imagen de CUs y TUs codificadas previamente de la misma imagen. Más específicamente, un codificador de vídeo puede intra-predecir una TU actual de una imagen usando un modo de intra-predicción particular.

Tras la cuantificación puede llevarse a cabo la codificación de entropía de los datos cuantificados, por ejemplo según la codificación de longitud variable adaptable al contenido (CAVLC), la codificación aritmética binaria adaptable al contexto (CABAC), la codificación de entropía mediante división en intervalos de probabilidad (PIPE) u otra metodología de codificación de entropía. Una unidad de procesamiento configurada para la codificación de entropía, u otra unidad de procesamiento, puede llevar a cabo otras funciones de procesamiento, tal como la codificación por longitud de series de cero de coeficientes cuantificados y/o la generación de información sintáctica, tal como valores de patrón de bloque codificado (CBP), el tipo de macrobloque, el modo de codificación, el tamaño máximo de macrobloque para una unidad codificada (tal como una trama, un fragmento, un macrobloque o una secuencia), u otra información de sintaxis.

El codificador de vídeo 20 puede configurarse para llevar a cabo una cuantificación inversa y una transformación inversa para almacenar bloques decodificados que van a usarse como referencia para predecir bloques posteriores, por ejemplo en la misma trama o tramas que van a predecirse de manera temporal. Además, el codificador de vídeo 5
20 puede enviar datos sintácticos, tales como datos sintácticos basados en bloques, datos sintácticos basados en tramas y datos sintácticos basados en GOP, al decodificador de vídeo 30, por ejemplo, en una cabecera de trama, una cabecera de bloque, una cabecera de fragmento o una cabecera de GOP. Los datos sintácticos de GOP pueden describir un cierto número de tramas en el GOP respectivo, y los datos sintácticos de trama pueden indicar un modo de codificación/predicción usado para codificar la trama correspondiente.

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden implementarse como cualquiera entre una amplia variedad de sistemas de circuitos codificadores o decodificadores adecuados, según corresponda, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), matrices de compuertas de campo programable (FPGA), circuitos de lógica discreta, software, hardware, firmware o cualquier combinación de los mismos. Cada uno, entre el codificador de vídeo 20 y el decodificador de vídeo 30, puede estar incluido en uno o más codificadores o decodificadores, cada uno de los cuales puede estar integrado como parte de un codificador/decodificador (CODEC) de vídeo combinado. Un aparato que incluye un codificador de vídeo 20 y/o un decodificador de vídeo 30 puede comprender un circuito integrado, un microprocesador y/o un dispositivo de comunicación inalámbrico, tal como un teléfono celular.

De acuerdo con las técnicas de esta divulgación, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden configurarse para almacenar una o más unidades de decodificación de datos de vídeo en una memoria intermedia de imágenes, obtener un tiempo de eliminación de memoria intermedia respectivo para las una o más unidades de decodificación, eliminar las unidades de decodificación de la memoria intermedia de imágenes de acuerdo con el tiempo de eliminación de memoria intermedia obtenido para cada una de las unidades de decodificación, y codificar datos de vídeo correspondientes a las unidades de decodificación eliminadas, entre otras funciones.

Las siguientes definiciones se proporcionan con respecto al codificador de vídeo 20 ejemplar y/o el decodificador de vídeo 30 configurados para almacenar una o más unidades de decodificación de datos de vídeo en una memoria intermedia de imágenes, obtener un tiempo de eliminación de memoria intermedia respectivo para las una o más unidades de decodificación, eliminar las unidades de decodificación de la memoria intermedia de imágenes de acuerdo con el tiempo de eliminación de memoria intermedia obtenido para cada una de las unidades de decodificación, y codificar datos de vídeo correspondientes a las unidades de decodificación eliminadas, entre otras funciones.

Para los fines de la descripción de un conjunto de ejemplos, la expresión "unidad de decodificación" se puede definir como se indica a continuación:

Unidad de decodificación: una unidad de acceso o un subconjunto de una unidad de acceso. Si `sub_pic_cpb_flag` es igual a 0, una unidad de decodificación es una unidad de acceso; de otro modo, una unidad de decodificación, es un subconjunto de una unidad de acceso. Cuando `sub_pic_cpb_flag` es igual a 1, la primera unidad de decodificación en una unidad de acceso incluye todas las unidades NAL no VCL en la unidad de acceso y la primera unidad NAL VCL en la misma unidad de acceso, y cada otra unidad de decodificación en la unidad de acceso es una unidad NAL de porción codificada que no es la primera unidad NAL de porción codificada en la unidad de acceso.

Para fines de descripción de un segundo conjunto de ejemplos, la expresión "unidad de decodificación" se puede definir como se indica a continuación, con una definición adicional del término "subimagen", como se usa en la definición ejemplar correspondiente de la "unidad de decodificación":

Unidad de decodificación: una unidad de acceso o un subconjunto de una unidad de acceso. Si `sub_pic_cpb_flag` es igual a 0, una unidad de decodificación es una unidad de acceso; de otro modo, una unidad de decodificación, es un subconjunto de una unidad de acceso. Cuando `sub_pic_cpb_flag` es igual a 1, la primera unidad de decodificación en una unidad de acceso incluye todas las unidades NAL no VCL en la unidad de acceso y la primera subimagen de la imagen en la misma unidad de acceso, y cada otra unidad de decodificación en la unidad de acceso es una subimagen que no es la primera subimagen en la unidad de acceso.

Sub-imagen: Varios bloques de una imagen codificada continua en orden de decodificación.

En la definición de acuerdo con el segundo conjunto de ejemplos proporcionados anteriormente, cuando se incluye más de una subimagen en una porción, la alineación de bytes se puede aplicar al final de cada subimagen, por el contrario, por ejemplo, de la alineación de bytes para las piezas en HEVC WD6. Además, se puede señalar el punto de entrada de cada subimagen, excepto para el primero en la imagen codificada.

En algunas alternativas, cuando el flujo de bits contiene múltiples capas o vistas escalables, una unidad de decodificación puede ser definida como una representación de capa o un componente de vista. Todas las unidades

no VCL anteriores a la primera unidad NAL VCL de una representación de capa o un componente de vista también pertenecen a la unidad de decodificación que contiene la representación de capa o el componente de vista.

Las siguientes descripciones del funcionamiento del decodificador de referencia hipotético (HRD) ejemplar, la operación ejemplar de una memoria intermedia de imagen codificada, la temporización ejemplar de una llegada de flujo de bits, la temporización ejemplar de la eliminación de la unidad de decodificación, la decodificación ejemplar de una unidad de decodificación, la operación ejemplar de una memoria intermedia de imágenes decodificadas, la eliminación ejemplar de las imágenes de una memoria intermedia de imágenes decodificadas, la salida de imagen ejemplar, y el marcado y almacenamiento ejemplar de imagen decodificada actual se proporcionan para ilustrar ejemplos del codificador de vídeo 20 y/o el decodificador de vídeo 30 que pueden configurarse para almacenar una o más de decodificación de unidades de datos de vídeo en una memoria intermedia de imágenes, obtener un tiempo de eliminación de memoria intermedia respectivo para las una o más unidades de decodificación, eliminar las unidades de decodificación de la memoria intermedia de imágenes de acuerdo con el tiempo de eliminación de memoria intermedia obtenido para cada una de las unidades de decodificación, y codificar los datos de vídeo correspondientes a las unidades de decodificación eliminadas, entre otras funciones. Las operaciones pueden definirse o realizarse de manera diferente, en otros ejemplos. De esta manera, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden estar configurados para funcionar de acuerdo con los diversos ejemplos de operaciones HRD que se describen a continuación.

Como un ejemplo, una operación HRD se puede describir en resumen como se indica a continuación: el tamaño CPB (número de bits) es $CpbSize[SchedSelIdx]$. El tamaño DPB (número de memorias intermedias de almacenamiento de imágenes) para la capa temporal X es $max_dec_pic_buffering[X] + 1$ para cada X en el intervalo de 0 a $max_temporal_layers_minus1$, inclusive. En este ejemplo, el HRD puede operar como se indica a continuación: los datos asociados a las unidades de acceso que fluyen en la CPB de acuerdo con una programación de llegada especificado pueden ser entregarse por el programador de flujo hipotético (HSS), es decir, un programador de la entrega. Los datos asociados a cada unidad de decodificación pueden eliminarse y decodificarse instantáneamente por el proceso de decodificación instantánea en los tiempos de eliminación de CPB. Cada imagen decodificada puede colocarse en la DPB. Una imagen decodificada puede eliminarse de la DPB por último en el tiempo de salida de DPB o el momento en que ya no sea necesario para una referencia inter-predicción.

La aritmética en estos ejemplos se puede hacer con valores reales, de modo que no se propaguen errores de redondeo. Por ejemplo, el número de bits en una CPB justo antes o después de la eliminación de una unidad de decodificación no es necesariamente un número entero.

La variable t_c se puede derivar como se indica a continuación y puede denominarse un pulso de reloj:

$$t_c = num_units_in_tick \div time_scale \quad (C-1)$$

Lo siguiente puede especificarse para expresar las limitaciones en una modificación anexo ejemplar de HEVC:

dejar que la unidad de acceso n sea la enésima unidad de acceso en el orden de decodificación siendo la primera unidad de acceso una unidad de acceso 0;

dejar que la imagen n sea la imagen codificada o la imagen decodificada de la unidad de acceso n;

dejar que la unidad de decodificación m sea la m-ésima unidad de decodificación en el orden de decodificación siendo la primera unidad de decodificación una unidad de decodificación 0.

Algunas técnicas de ejemplo para la operación de una memoria intermedia de imagen codificada (CPB) se describen como se indica a continuación. De acuerdo con algunas técnicas de codificación de vídeo, pueden implementarse diversos procedimientos de operación CPB. Las especificaciones en la sección de HEVC WD6 en las operaciones del CPB pueden ser modificados por esta divulgación, y se pueden aplicar de forma independiente a cada conjunto de parámetros CPB que está presente y a los puntos de conformidad tanto de Tipo I como de tipo II.

Algunos ejemplos que implican la temporización de llegada de flujo de bits se describen como se indica a continuación. El HRD se puede inicializar en cualquiera de los mensajes de información de mejora complementaria del período de almacenamiento en memoria intermedia. Antes de la inicialización, la CPB puede estar vacía. Después de la inicialización, el HRD puede no inicializarse de nuevo por los posteriores mensajes SEI del período de almacenamiento intermedio.

La unidad de acceso que se asocia al mensaje SEI del período de almacenamiento intermedio que inicializa la CPB puede denominarse como la unidad de acceso 0. Cada unidad de decodificación puede denominarse como una unidad de decodificación m, donde el número m identifica la unidad de decodificación particular. La primera unidad de decodificación en orden de decodificación en la unidad de acceso 0 puede denominarse como la unidad de decodificación 0. El valor de m puede aumentarse en 1 para cada unidad de decodificación posterior en orden de decodificación.

El momento en el que el primer bit de la unidad de decodificación m comienza a entrar en la CPB puede denominarse como el tiempo de llegada inicial $t_{ai}(m)$. La hora de llegada inicial de las unidades de decodificación puede obtenerse como se indica a continuación:

5 si la unidad de decodificación es la unidad de decodificación 0, $t_{ai}(0) = 0$, de otro modo (la unidad de decodificación en la unidad de decodificación m con $m > 0$), puede aplicarse lo siguiente:

10 si $cbr_flag[SchedSelIdx]$ es igual a 1, el tiempo de llegada inicial para la unidad de decodificación m, es igual al tiempo final de la llegada (que se obtiene a continuación) de la unidad de decodificación m - 1, es decir,

$$t_{ai}(m) = t_{af}(m - 1) \quad (C-2)$$

15 de otro modo ($cbr_flag[SchedSelIdx]$ es igual a 0), el tiempo de llegada inicial de la unidad de decodificación m se obtiene por

$$t_{ai}(m) = \text{Max}(t_{af}(m - 1), t_{ai,m\acute{a}stemprano}(m)) \quad (C-3)$$

20 donde $t_{ai,m\acute{a}stemprano}(m)$ se obtiene como se indica a continuación:

Si la unidad de decodificación m no es la primera unidad de decodificación de un período de almacenamiento en memoria intermedia posterior, $t_{ai,m\acute{a}stemprano}(m)$ puede obtenerse como:

$$25 \quad t_{ai,m\acute{a}stemprano}(m) = t_{r,n}(m) - (\text{initial_cpb_removal_delay}[SchedSelIdx] + \text{initial_cpb_removal_delay_offset}[SchedSelIdx]) \div 90000 \quad (C-4)$$

30 siendo $t_{r,n}(m)$ el tiempo de eliminación nominal de la unidad de decodificación m de la CPB como se especifica e $\text{initial_cpb_removal_delay}[SchedSelIdx]$ e $\text{initial_cpb_removal_delay_offset}[SchedSelIdx]$ especificándose en el mensaje SEI del periodo de almacenamiento intermedio anterior;

de otro modo (la unidad de decodificación n no es la primera unidad de decodificación de un período de almacenamiento en memoria intermedia posterior), $t_{ai,m\acute{a}stemprano}(m)$ puede obtenerse como

$$35 \quad t_{ai,m\acute{a}stemprano}(m) = t_{r,n}(m) - (\text{initial_cpb_removal_delay}[SchedSelIdx] \div 90000) \quad (C-5)$$

estando $\text{initial_cpb_removal_delay}[SchedSelIdx]$ especificado en el mensaje SEI del periodo de almacenamiento intermedio asociado a la unidad de acceso que contiene la unidad de decodificación m.

40 El tiempo de llegada final para la unidad de decodificación m puede obtenerse por:

$$t_{af}(m) = t_{ai}(m) + b(m) \div \text{BitRate}[SchedSelIdx] \quad (C-6)$$

45 donde $b(m)$ puede ser el tamaño en bits de la unidad de decodificación m, contando los bits de las unidades NAL VCL y las unidades NAL de datos de relleno para el punto de conformidad de Tipo I o todos los bits del flujo de bits de Tipo II para el punto de la conformidad de Tipo II.

50 En algunos ejemplos, los valores de $SchedSelIdx$, $\text{BitRate}[SchedSelIdx]$, y $\text{CpbSize}[SchedSelIdx]$ pueden limitarse como se indica a continuación:

Si el contenido de los conjuntos de parámetros de secuencia activa para la unidad de acceso que contiene la unidad de decodificación m y la unidad de acceso anterior difieren, el HSS selecciona un valor $SchedSelIdx1$ de $SchedSelIdx$ de entre los valores de $SchedSelIdx$ proporcionados en el conjunto de parámetros de secuencia activa para la unidad de acceso que contiene la unidad de decodificación m que se traduce en $\text{BitRate}[SchedSelIdx1]$ o $\text{CpbSize}[SchedSelIdx1]$ para la unidad de acceso que contiene la unidad de decodificación m. El valor de $\text{BitRate}[SchedSelIdx1]$ o $\text{CpbSize}[SchedSelIdx1]$ puede diferir del valor de $\text{BitRate}[SchedSelIdx0]$ o $\text{CpbSize}[SchedSelIdx0]$ para el valor de $SchedSelIdx0$ de $SchedSelIdx$ que estaba en uso para la unidad de acceso anterior;

60 de otro modo, el HSS sigue funcionando con los valores anteriores de $SchedSelIdx$, $\text{BitRate}[SchedSelIdx]$ y $\text{CpbSize}[SchedSelIdx]$.

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

65 la variable $\text{BitRate}[SchedSelIdx]$ entra en vigor en el momento $t_{ai}(m)$ la variable de $\text{CpbSize}[SchedSelIdx]$ entra

en vigor como se indica a continuación:

5 si el nuevo valor de $CpbSize[SchedSelIdx]$ excede el tamaño de CPB anterior, entra en vigor en el momento $t_{ai}(m)$, de otro modo, el nuevo valor de $CpbSize[SchedSelIdx]$ entra en vigor en el momento de la eliminación de la CPB de la última unidad de decodificación de la unidad de acceso que contiene la unidad de decodificación m .

10 Cuando $sub_pic_cpb_flag$ es igual a 1, el tiempo de llegada de CPB inicial de la unidad de acceso n $t_{ai}(n)$ puede ajustarse en el tiempo de llegada de CPB inicial de la primera unidad de decodificación en la unidad de acceso n , y el tiempo de llegada de CPB final de la unidad de acceso n $t_{af}(n)$ puede ajustarse en el tiempo de llegada de CPB final de la última unidad de decodificación en la unidad de acceso n .

15 Algunos ejemplos que implican la temporización de la eliminación de la unidad de decodificación y la decodificación de una unidad de decodificación se describen como se indica a continuación. Cuando una unidad de decodificación m es la unidad de decodificación con m igual a 0 (la primera unidad de decodificación de la unidad de acceso que inicializa el HRD), el tiempo de eliminación nominal de la unidad de decodificación de la CPB puede especificarse por:

$$20 \quad t_{r,n}(0) = initial_cpb_removal_delay[SchedSelIdx] \div 90000 \quad (C-7)$$

25 Cuando una unidad de decodificación m es la primera unidad de decodificación de la primera unidad de acceso de un período de almacenamiento en memoria intermedia que no inicializa el HRD, el tiempo de eliminación nominal de la unidad de decodificación de la CPB puede especificarse por:

$$25 \quad t_{r,n}(m) = t_{r,n}(m_b) + t_c * cpb_removal_delay(m) \quad (C-8)$$

30 donde $t_{r,n}(m_b)$ es el tiempo de eliminación nominal de la primera unidad de decodificación del período de almacenamiento intermedio anterior y $cpb_removal_delay(m)$ es el valor de $cpb_removal_delay[i]$ para la unidad de decodificación m que se especifica en el mensaje SEI de temporización de imagen asociado a la unidad de acceso que contiene la unidad de decodificación m .

35 Cuando una unidad de decodificación n es la primera unidad de decodificación de un período de almacenamiento en memoria intermedia, m_b puede ajustarse igual a m en el momento de la eliminación $t_{r,n}(m)$ de la unidad de decodificación n . El tiempo de eliminación nominal $t_{r,n}(m)$ de una unidad de decodificación m que no es la primera unidad de decodificación de un período de almacenamiento en memoria intermedia puede darse por:

$$35 \quad t_{r,n}(m) = t_{r,n}(m_b) + t_c * cpb_removal_delay(m) \quad (C-9)$$

40 donde $t_{r,n}(m_b)$ es el tiempo de eliminación nominal de la primera unidad de decodificación del período de almacenamiento intermedio actual y $cpb_removal_delay(m)$ es el valor de $cpb_removal_delay[i]$ para la unidad de decodificación m que se especifica en el mensaje SEI de temporización de imagen asociado a la unidad de acceso que contiene la unidad de decodificación m .

45 La hora de eliminación de la unidad de decodificación m puede especificarse como se indica a continuación:

si $low_delay_hrd_flag$ es igual a 0 o $t_{r,n}(m) \geq t_{af}(m)$, el tiempo de eliminación de la unidad de decodificación n puede especificarse por:

$$50 \quad t_r(m) = t_{r,n}(m) \quad (C-10)$$

de otro modo ($low_delay_hrd_flag$ es igual a 1 y $t_{r,n}(m) < t_{af}(m)$), el tiempo de eliminación de la unidad de decodificación m se especifica por:

$$55 \quad t_r(m) = t_{r,n}(m) + t_c * Ceil((t_{af}(m) - t_{r,n}(m)) \div t_c) \quad (C-11)$$

El último caso indica que el tamaño de la unidad de decodificación m , $b(m)$, es tan grande que se impide la eliminación en el tiempo de eliminación nominal.

60 Cuando $sub_pic_cpb_flag$ es igual a 1, el tiempo de eliminación de CPB nominal de la unidad de acceso n $t_{r,n}(n)$ puede ajustarse en el tiempo de eliminación de CPB normal de la última unidad de decodificación en la unidad de acceso n , el tiempo de eliminación de CPB de la unidad de acceso n $t_r(n)$ puede ajustarse en el tiempo de eliminación de CPB de la última unidad de decodificación en la unidad de acceso n .

65 En algunos ejemplos, en el tiempo de eliminación de CPB de la unidad de decodificación m , la unidad de decodificación puede decodificarse instantáneamente.

Algunos ejemplos de operación de la memoria intermedia de imagen decodificada (DPB) se describen como se indica a continuación. La memoria intermedia de imágenes decodificadas puede contener memorias intermedias de almacenamiento de imágenes. Cada una de las memorias intermedias de almacenamiento de imágenes puede contener una imagen decodificada que se marca como "usada para referencia" o se guarda para una salida futura. Antes del inicio, la DPB puede estar vacía (la plenitud de la DPB se ajusta a cero). Las siguientes etapas de estos ejemplos de las técnicas de esta divulgación pueden suceder en la secuencia indicada.

Algunos ejemplos de eliminación de imágenes de la memoria intermedia de imagen decodificada (DPB) se describen como se indica a continuación. En algunos ejemplos, la eliminación de imágenes de la DPB antes de la decodificación de la imagen actual (pero después de analizar el encabezado de fragmento del primer fragmento de la imagen actual) puede tener lugar instantáneamente en el momento de la eliminación de la CPB de la primera unidad de decodificación de la unidad de acceso n (que contiene la imagen actual) y puede avanzar como se indica a continuación.

Puede invocarse el proceso de decodificación para la imagen de referencia ajustada como se especifica en la subcláusula 8.3.2 de HEVC WD6. Si la imagen actual es una imagen de regeneración de decodificador instantáneo (IDR), puede aplicarse lo siguiente:

1. Cuando la imagen IDR no es la primera imagen IDR decodificada (por ejemplo, cuando ningún resultado del indicador de las imágenes anteriores tiene un valor no igual a 1) y el valor de `pic_width_in_luma_samples` (por ejemplo, un ancho de imagen en las muestras luma) `pic_height_in_luma_samples` o `max_dec_pic_buffering` que se obtiene a partir del conjunto de parámetros de secuencia activa es diferente del valor de `pic_width_in_luma_samples` o `pic_height_in_luma_samples` o `max_dec_pic_buffering` obtenido a partir del conjunto de parámetro de secuencia que era activo para la imagen anterior, respectivamente, `no_output_of_prior_pics_flag` puede inferirse para ser igual a 1 o ajustarse que sea igual a 1 por el HRD, independientemente del valor real de `no_output_of_prior_pics_flag`. Las implementaciones del decodificador pueden manipular los cambios en la imagen o del tamaño de DPB más fácilmente que el HRD con respecto a cambios en `pic_width_in_luma_samples` o `pic_height_in_luma_samples`.

2. Cuando `no_output_of_prior_pics_flag` es igual a 1 o se ajusta a 0 o se deduce que es igual a 1, todas las memorias intermedias de almacenamiento de imagen en la DPB pueden estar vacías sin salida de las imágenes que contienen, y la plenitud de DPB se puede ajustar a 0.

Todas las imágenes k en la DPB, para las cuales son ciertas cada una de las siguientes condiciones, pueden eliminarse de la DPB: la imagen k se marca como "no utilizada como referencia"; la imagen k tiene un `PicOutputFlag` igual a 0 o su tiempo de salida de DPB es menor que o igual al tiempo de eliminación de CPB de la primera unidad de decodificación (representada como la unidad de decodificación m) de la imagen actual n; es decir,

$$t_{o,dpb}(k) \leq t_r(m)$$

Cuando una imagen se elimina de la DPB, la plenitud de la DPB puede reducirse en uno.

Algunos ejemplos de salida de imagen se describen como se indica a continuación. Lo siguiente puede suceder instantáneamente en el tiempo de eliminación de CPB de la última unidad de decodificación (representada como unidad de decodificación m) de la unidad de acceso n (que contiene la imagen actual), $t_r(m)$. La imagen n puede considerarse como decodificada después de que se decodifique la última unidad de decodificación de la imagen.

La variable `maxPicOrderCnt` variable (para el recuento de orden de imagen máximo (POC)) puede ajustarse igual al máximo de los valores `PicOrderCntVal` (para el valor de recuento de orden de imágenes (POC)) para la imagen actual y todas las imágenes en la DPB que están marcadas actualmente como "usadas como referencia a corto plazo", o que tienen un tiempo de salida de DPB mayor de $t_r(m)$. La variable `minPicOrderCnt` (para el recuento de orden de imagen mínimo (POC), es decir, el recuento de orden de imagen menor (POC)) puede ajustarse igual al mínimo del valor `PicOrderCntVal` para la imagen actual y todas las imágenes en la DPB que están marcadas actualmente como "usadas como referencia a corto plazo", o que tienen un tiempo de salida de DPB mayor de $t_r(m)$. Puede ser un requisito de la conformidad del flujo de bits que el valor de `maxPicOrderCnt` – `minPicOrderCnt` sea menor de `MaxPicOrderCntLsb` / 2.

Cuando la imagen n tiene `PicOutputFlag` igual a 1, su tiempo de salida de DPB $t_{o,dpb}(n)$ puede obtenerse por:

$$t_{o,dpb}(n) = t_r(m) + t_c * dpb_output_delay(n) \tag{C-12}$$

donde `dpb_output_delay(n)` es el valor de `dpb_output_delay` especificado en el mensaje de SEI de temporización de imagen asociado a la unidad de acceso n. La salida de la imagen actual puede especificarse como se indica a continuación:

si PicOutputFlag es igual a 1 y $t_{o,dpb}(n) = tr(m)$, la imagen actual se transmite;

de otro modo, PicOutputFlag es igual a 0, la imagen actual no se emite, sino que puede almacenarse en la DPB como se especifica adicionalmente a continuación;

de otro modo (PicOutputFlag es igual a 1 y $t_{o,dpb}(n) > tr(m)$), la imagen actual se transmite más tarde y se almacenará en la DPB (como se especifica adicionalmente a continuación) y se transmite en el momento $t_{o,dpb}(n)$, a menos que se indique que no se transmite por la decodificación o inferencia de no_output_of_prior_pics_flag igual a 1 en un momento que antecede a $t_{o,dpb}(n)$.

Al transmitirse, la imagen actual o seleccionada puede recortarse, usando un rectángulo de recorte especificado en el conjunto de parámetros de secuencia activa, generando de este modo una imagen recortada en base a la imagen seleccionada, es decir, la imagen actual. Cuando la imagen n es una imagen que se emite y no es la última imagen del flujo de bits que se emite, el valor de $\Delta t_{o,dpb}(n)$ se define como:

$$\Delta t_{o,dpb}(n) = t_{o,dpb}(n_n) - t_{o,dpb}(n) \tag{C-13}$$

donde n_n indica la imagen que sigue a la imagen n en orden de salida y tiene PicOutputFlag igual a 1. A continuación, se proporcionan adicionalmente más detalles de un proceso de abultamiento y un proceso de recorte.

Algunos ejemplos que implican el marcado y el almacenamiento de la imagen decodificada actual se describen como se indica a continuación. Lo siguiente puede suceder instantáneamente en el tiempo de eliminación de CPB de la última unidad de decodificación de la unidad de acceso n (que contiene la imagen actual), $t_r(m)$. La imagen decodificada actual puede almacenarse en la DPB en una memoria de almacenamiento de imagen vacía, y la plenitud de la DPB puede aumentarse en uno. Si la imagen actual es una imagen de referencia, puede ser marcada como "utilizada como referencia"; de lo contrario se puede marcar como "no utilizado como referencia".

La siguiente sintaxis y semántica ejemplar para señalar el modo de comportamiento de la CPB se proporcionan con respecto al codificador de vídeo ejemplar 20 y/o el decodificador de vídeo 30 configurados para almacenar una o más unidades de decodificación de datos de vídeo en una memoria intermedia de imágenes, obtener un tiempo de eliminación de memoria intermedia respectivo para las una o más unidades de decodificación, eliminar las unidades de decodificación de la memoria intermedia de imágenes de acuerdo con el tiempo de eliminación de memoria intermedia obtenido para cada una de las unidades de decodificación, y codificar datos de vídeo correspondientes a las unidades de decodificación eliminadas, entre otras funciones.

Algunos ejemplos de sintaxis y semántica para la señalización del modo de comportamiento de la CPB se describen como se indica a continuación. La sintaxis de los parámetros de información de usabilidad de vídeo (VUI) puede cambiarse mediante la adición de un indicador de CPB de subimagen, sub_pic_cpb_flag, como se muestra en la Tabla 1 a continuación:

Tabla 1

vui_parameters() {	Descriptor
...	
timing_info_present_flag	u(1)
if(timing_info_present_flag) {	
num_units_in_tick	u(32)
Time_scale	u(32)
fixed_pic_rate_flag	u(1)
}	
nal_hrd_parameters_present_flag	u(1)
if(nal_hrd_parameters_present_flag)	
hrd_parameters()	
vcl_hrd_parameters_present_flag	u(1)
if(vcl_hrd_parameters_present_flag)	
hrd_parameters()	
if(nal_hrd_parameters_present_flag) vcl_hrd_parameters_present_flag) {	
sub_pic_cpb_flag	u(1)
lowdelayhrdflag	u(1)
}	
...	
}	

En este ejemplo, la Tabla 1 incluye un indicador añadido "sub_pic_cpb_flag", con respecto al HEVC convencional.

Este indicador de CPB de subimagen, "sub_pic_cpb_flag", puede usarse para indicar si un conjunto de datos de vídeo proporcionados a la memoria intermedia de imágenes codificación (CPB) incluye o no parámetros de subimagen para la decodificación de subimágenes. Tales parámetros de subimagen de los cuales el indicador "sub_pic_cpb_flag" pueden indicar la presencia pueden incluir los tiempos de eliminación de la memoria intermedia, incluyendo tiempos de eliminación de memoria intermedia respectivos (es decir, tiempos de eliminación de CPB) para cada una de una o más unidades de decodificación. Un ejemplo de la semántica de sub_pic_cpb_flag es como se indica a continuación. El elemento de sintaxis sub_pic_cpb_flag que es igual a 0 puede especificar que la CPB opera a nivel de la unidad de acceso. El elemento de sintaxis sub_pic_cpb_flag igual a 1 puede especificar que la CPB funciona a un nivel de la unidad de decodificación que puede ser a nivel de las unidades de acceso o de subconjuntos de unidades de acceso, lo que puede corresponder a las subimágenes. Cuando el indicador sub_pic_cpb no está presente, su valor puede ajustarse o deducirse como igual a 0, lo que puede indicar un estado predeterminado en el que los datos de vídeo no incluyen parámetros de subimagen para la decodificación de subimagen.

Algunos ejemplos de sintaxis y semántica para la señalización de los tiempos de eliminación de CPB de las unidades de decodificación se describen como se indica a continuación. La sintaxis del mensaje SEI del periodo de almacenamiento intermedio puede permanecer inalterada, al igual que en HEVC WD6, mientras que la semántica de los elementos de sintaxis initial_cpb_removal_delay[SchedSelIdx] e initial_cpb_removal_delay_offset[SchedSelIdx] pueden cambiarse como se indica a continuación. En este ejemplo, el elemento de sintaxis initial_cpb_removal_delay[SchedSelIdx] puede especificar el retardo para la SchedSelIdx-ésima CPB entre el momento de llegada a la CPB del primer bit de los datos codificados asociados a la primera unidad de decodificación en la unidad de acceso asociada al mensaje SEI del periodo de almacenamiento intermedio y el momento de la eliminación de la CPB de los datos codificados asociados a la misma unidad de decodificación, para el primer período de memoria intermedia después de la inicialización HRD. Este elemento sintáctico puede tener una longitud en bits dada por initial_cpb_removal_delay_length_minus1 + 1. Esto puede referirse a unidades de un reloj de 90 kHz. En este ejemplo, este elemento de sintaxis initial_cpb_removal_delay [SchedSelIdx] no puede ser igual a 0 y no podrá ser superior a $90000 * (CpbSize [SchedSelIdx] \div bitrate [SchedSelIdx])$, el tiempo equivalente del tamaño de la CPB en unidades de reloj de 90 kHz.

En este ejemplo, el elemento de sintaxis initial_cpb_removal_delay_offset [SchedSelIdx] puede ser utilizado para la SchedSelIdx-ésima CPB en combinación con el elemento de sintaxis cpb_removal_delay para especificar el tiempo de entrega inicial de las unidades de decodificación a la CPB. Además, el elemento de sintaxis initial_cpb_removal_delay_offset[SchedSelIdx] puede ser en unidades de un reloj de 90 kHz. El elemento de sintaxis initial_cpb_removal_delay_offset[SchedSelIdx] puede ser un código de longitud fija cuya longitud en bits se da por initial_cpb_removal_delay_length_minus1 + 1. Este elemento de sintaxis puede no usarse por los decodificadores y puede ser necesario sólo para el programador de entrega (HSS) especificado en el Anexo C de HEVC WD6.

En algunos ejemplos, la sintaxis y la semántica del mensaje SEI de temporización de imagen se pueden cambiar como se muestra en la Tabla 2 a continuación:

Tabla 2

	Descriptor
pic_timing(payloadSize) {	
if(CpbDpbDelaysPresentFlag) {	
if(sub_pic_cpb_flag)	
num_decoding_units_minus1	ue(v)
for(i = 0; i <= numdecodingunitsminus1;	
cpb_removal_delay[i]	u(v)
dpb_output_delay	u(v)
}	
}	
}	

En el ejemplo de la Tabla 2, el mensaje SEI pictiming incluye una señal añadida num_decoding_units_minus1, y un bucle sobre el número de unidades de decodificación, lo que indica un retraso de eliminación respectivo de una unidad de decodificación de la memoria intermedia de imagen codificada, cuando el sub_pic_cpb_flag de los parámetros VUI, por ejemplo, de acuerdo con la Tabla 1 anterior, es cierto. De esta manera, el mensaje SEI pic_timing puede incluir información que indica un número de ciclos de reloj que esperar hasta la eliminación de cada una de una pluralidad de unidades de decodificación de la memoria intermedia de imagen codificada cuando los parámetros VUI indican que la CPB opera a nivel de la unidad de decodificación. El retardo de la eliminación de una unidad de decodificación puede ser el mismo retardo para cada unidad de decodificación en una carga útil u otra unidad de datos. En otros ejemplos, se pueden aplicar diferentes retardos de eliminación a diferentes unidades de decodificación. El retardo de eliminación puede ser expresado en cuando a un número de bits, con una conversión de tiempo implícito del número de bits con respecto a la tasa de procesamiento de bits para el reloj aplicable.

La sintaxis del mensaje SEI de temporización de imagen puede depender del contenido del conjunto de parámetros

de secuencia que es activo para la imagen codificada asociada al mensaje SEI de temporización de imagen. Sin embargo, a menos que el mensaje SEI de temporización de imagen de una unidad de acceso de regeneración de decodificación instantánea (IDR) está precedido de un mensaje SEI del período de almacenamiento intermedio dentro de la misma unidad de acceso, la activación del conjunto de parámetros de secuencia asociado (y, para imágenes IDR que no son la primera imagen en el flujo de bits, la determinación de que la imagen codificada es una imagen IDR) puede no producirse hasta la decodificación de la primera unidad de capa de abstracción de red (NAL) de porción codificada de la imagen codificada. Dado que la unidad NAL de porción codificada de la imagen codificada sigue al mensaje SEI de temporización de imagen en el orden de la unidad NAL, puede haber casos en los que es necesario para un decodificador almacenar la carga útil de secuencia de bytes en bruto (RBSP) que contiene el mensaje SEI de temporización de imagen hasta la determinación de los parámetros del conjunto de parámetros de secuencia que estarán activos para la imagen codificada, y después realizar el análisis del mensaje SEI de temporización de imagen. El decodificador puede almacenar una o más unidades de decodificación de datos de vídeo en un orden de decodificación continuo en la memoria intermedia de imagen.

La presencia del mensaje SEI de temporización de imagen en el flujo de bits puede especificarse en un ejemplo como se indica a continuación: si `CpbDpbDelaysPresentFlag` es igual a 1, un mensaje SEI de temporización de imagen puede estar presente en cada unidad de acceso de la secuencia de vídeo codificada. De otro modo, `CpbDpbDelaysPresentFlag` es igual a 0, y ningún mensaje SEI de temporización de imagen puede estar presente en ninguna unidad de acceso de la secuencia de vídeo codificada.

En este ejemplo, el elemento de sintaxis `num_decoding_units_minus1` más 1 puede especificar el número de unidades de decodificación en la unidad de acceso a la que está asociado mensaje SEI de temporización de imagen. Cuando `sub_pic_cpb_flag` es igual a 0, el elemento de sintaxis `numdecodingunitsminus1` puede no estar presente y el valor puede ajustarse o deducirse como 0.

En este ejemplo, el elemento de sintaxis `cpb_removal_delay[i]` puede especificar el número de ciclos de reloj a esperar después de la eliminación de la CPB de la primera unidad de decodificación en la unidad de acceso asociada al mensaje SEI del período de almacenamiento intermedio más reciente en una unidad de acceso anterior antes de la eliminación de la CPB de la *i*-ésima unidad de decodificación en la unidad de acceso asociada al mensaje SEI de temporización de imagen. Este valor también puede usarse para calcular un tiempo lo más temprano posible de llegada de datos de la unidad de decodificación a la CPB para el HSS. El elemento de sintaxis puede ser un código de longitud fija, cuya longitud en bits se da por `cpb_removal_delay_length_minus1 + 1`. El `cpb_removal_delay[i]` puede ser el resto de un contador de módulo 2^(`cpb_removal_delay_length_minus1+1`).

El valor de `cpb_removal_delay_length_minus1` que determina la longitud (en bits) del elemento de sintaxis `cpb_removal_delay [i]` puede ser el valor de `cpb_removal_delay_length_minus1` codificado en el conjunto de parámetros de secuencia que está activo para la imagen codificada asociada al mensaje SEI de temporización de imagen. Sin embargo, `cpb_removal_delay [i]` puede especificar un número de ciclos de reloj en relación con el tiempo de eliminación de la primera unidad de decodificación en la unidad de acceso anterior que contiene un mensaje SEI del período de almacenamiento intermedio, que puede ser una unidad de acceso de una secuencia de vídeo codificado diferente.

En este ejemplo, el elemento de sintaxis `dpb_output_delay` puede usarse para calcular el tiempo de salida DPB de la imagen. El elemento de sintaxis `dpboutputdelay` puede especificar el número de ciclos de reloj a esperar después de la eliminación de la última unidad de decodificación de una unidad de acceso de la CPB antes de que la imagen decodificada se transmita desde la DPB.

Una imagen puede no eliminarse de la DPB en su tiempo de salida cuando todavía está marcada como "utilizada como referencia a corto plazo" o "utilizada como referencia a largo plazo". Sólo se puede especificar un `dpb_output_delay` para una imagen decodificada. La longitud del elemento de sintaxis `dpboutputdelay` se puede dar en bits por `dpb_output_delay_length_minus1 + 1`. Cuando `max_dec_pic_buffering[max_temporal_layers_minus1]` es igual a 0, `dpb_output_delay` puede ser igual a 0.

El tiempo de salida obtenido de `dpb_output_delay` de cualquier imagen que se transmite desde un tiempo de salida conforme al decodificador puede preceder al momento de salida obtenido de `dpb_output_delay` de todas las imágenes en cualquier secuencia de vídeo codificada posterior en el orden de decodificación. El orden de salida de imagen establecido por los valores de este elemento de sintaxis puede ser el mismo orden que se establece por los valores de `PicOrderCnt()`. Para las imágenes que no se transmiten por el proceso de "abultamiento" porque preceden, en orden de decodificación, a una imagen IDR con `no_output_of_prior_pics_flag` igual a 1 o que se deduce como igual a 1, los tiempos de salida obtenidos de `dpb_output_delay` pueden estar aumentando con el valor en aumento de `PicOrderCnt ()` con respecto a todas las imágenes dentro de la misma secuencia de vídeo codificada. En un ejemplo alternativo, un nuevo mensaje SEI, que puede denominarse un mensaje SEI de temporización de la unidad de decodificación, cada uno asociado a una unidad de decodificación, puede especificarse, para transmitir el retardo de eliminación de CPB para la unidad de decodificación asociada.

De esta manera, implementando cualquier combinación de las definiciones ejemplares, una operación de HRD

ejemplar, la operación ejemplar de una memoria intermedia de imagen codificada, la temporización ejemplar de una llegada de flujo de bits, la temporización ejemplar de la eliminación de la unidad de decodificación, la decodificación ejemplar de una unidad de decodificación, la operación ejemplar de una memoria intermedia de imágenes decodificadas, la eliminación ejemplar de las imágenes de una memoria intermedia de imágenes decodificadas, la salida de imagen ejemplar, y el marcado y almacenamiento ejemplar de imagen decodificada actual, y la sintaxis y semántica ejemplar para señalar el modo de comportamiento de CPB, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden configurarse para almacenar una o más de decodificación de unidades de datos de vídeo en una memoria intermedia de imágenes, obtener un tiempo de eliminación de memoria intermedia respectivo para las una o más unidades de decodificación, eliminar las unidades de decodificación de la memoria intermedia de imágenes de acuerdo con el tiempo de eliminación de memoria intermedia obtenido para cada una de las unidades de decodificación, y codificar los datos de vídeo correspondientes a las unidades de decodificación eliminadas, entre otras funciones.

Como una alternativa a las técnicas descritas anteriormente, una unidad de decodificación se puede definir como se indica a continuación: "Una unidad de acceso o un subconjunto de una unidad de acceso. Si SubPicCpbFlag es igual a 0, una unidad de decodificación es una unidad de acceso. De otro modo, una unidad de decodificación incluye una o más unidades NAL VCL y las unidades NAL no VCL asociadas en una unidad de acceso. Para la primera unidad NAL VCL en una unidad de acceso, las unidades NAL no VCL asociadas son todas las unidades NAL no VCL en la unidad de acceso y antes de la primera unidad NAL VCL y las unidades NAL de datos de relleno, en su caso, inmediatamente después de la primera unidad NAL no VCL. Para una unidad NAL VCL que no es la primera unidad NAL VCL en una unidad de acceso, las unidades NAL no VCL asociadas son las unidades NAL de datos de relleno, en su caso, inmediatamente después de la unidad NAL no VCL".

En este ejemplo, las operaciones del decodificador de referencia hipotético (HRD) pueden resumirse como se indica a continuación. El tamaño de CPB (número de bits) es CpbSize[SchedSelIdx]. El tamaño DPB (número de memorias intermedias de almacenamiento de imágenes) para la capa temporal X puede ser $\max_dec_pic_buffering[X] + 1$ para cada X en el intervalo de 0 a $\max_temporal_layers_minus1$, inclusive. Una variable SubPicCpbPreferredFlag puede usarse como un indicador preferido de memoria intermedia de imágenes codificadas de subimagen, y puede especificarse por medios externos, o cuando no se especifica por medios externos, ajustarse a 0. Un indicador presente de parámetros de imagen codificada de subimagen, sub_pic_cpb_params_present_flag, puede usarse para señalar si los parámetros necesarios para los subconjuntos de codificación de una o más unidades de acceso están disponibles. Un único indicador de memoria intermedia de imagen codificada de subimagen, SubPicCpbFlag, puede indicar si tanto el indicador preferido de memoria intermedia de imagen codificada de subimagen como el indicador presente de parámetros de imagen codificada de subimagen son positivos, o se ajustan a 1. Un codificador de vídeo puede utilizar este indicador de memoria intermedia de imagen codificada de subimagen, SubPicCpbFlag, para determinar si codificar unidades de acceso de datos de vídeo o codificar subconjuntos de una o más unidades de acceso, tales como subimágenes, de datos de vídeo, según los datos de vídeo se eliminan de la CPB.

La variable SubPicCpbFlag puede obtenerse como se indica a continuación:

```
SubPicCpbFlag = SubPicCpbPreferredFlag && sub_pic_cpb_params_present_flag
```

(C-1)

Si SubPicCpbFlag es igual a 0, la CPB puede operar a nivel de unidad de acceso, y cada unidad de decodificación puede ser una unidad de acceso. De otro modo, la CPB puede operar a nivel de subimagen, y cada unidad de decodificación puede ser un subconjunto de una unidad de acceso.

El decodificador de vídeo 30/108 puede determinar que la una o más unidades de decodificación comprenden unidades de acceso determinando que un indicador preferido de memoria intermedia de imagen codificada de subimagen (por ejemplo, SubPicCpbPreferredFlag) tiene un valor de cero o que un indicador presente de parámetros de memoria intermedia de imagen codificada de subimagen (por ejemplo, sub_pic_cpb_params_present_flag) tiene un valor de cero.

El HRD (por ejemplo, el codificador de vídeo 20 y/o el decodificador de vídeo 30) puede funcionar como se indica a continuación. Los datos asociados con las unidades de decodificación que fluyen en la CPB de acuerdo con una programación de llegada especificada pueden entregarse por el HSS. En un ejemplo, los datos asociados a cada unidad de decodificación pueden eliminarse y decodificarse instantáneamente por el proceso de decodificación instantánea en los tiempos de eliminación de CPB. Cada imagen decodificada puede colocarse en la DPB. Una imagen decodificada puede eliminarse de la DPB por último en el tiempo de salida de DPB o el momento en que ya no sea necesario para una referencia inter-predicción.

Las operaciones aritméticas descritas en esta divulgación se pueden hacer con valores reales, de modo que no se propaguen los errores de redondeo. Por ejemplo, el número de bits en un CPB justo antes o después de la eliminación de una unidad de decodificación puede no ser necesariamente un número entero.

La variable t_c se puede derivar como se indica a continuación y puede denominarse un pulso de reloj:

$$t_c = \text{num_units_in_tick} \div \text{time_scale} \quad (\text{C-1})$$

5 Lo siguiente puede especificarse para expresar las limitaciones en este ejemplo de las técnicas de esta divulgación:

dejar que la unidad de acceso n sea la n -ésima unidad de acceso en el orden de decodificación siendo la primera unidad de acceso una unidad de acceso 0;

10 dejar que la imagen n sea la imagen codificada o la imagen decodificada de la unidad de acceso n ;

dejar que la unidad de decodificación m sea la m -ésima unidad de decodificación en el orden de decodificación siendo la primera unidad de decodificación una unidad de decodificación 0.

15 Las operaciones de la memoria intermedia de imagen codificada (CPB) se pueden definir como se indica a continuación. Las especificaciones en este ejemplo se pueden aplicar de forma independiente a cada conjunto de parámetros CPB que está presente y a los puntos de conformidad tanto de Tipo I como de Tipo II.

20 Con respecto al tiempo de llegada del flujo de bits, 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. Después de la inicialización, el HRD puede no inicializarse de nuevo por los posteriores mensajes SEI del período de almacenamiento intermedio.

25 Cada unidad de acceso puede denominarse como una unidad de acceso respectiva n , donde el número n identifica la unidad de acceso particular. La unidad de acceso que se asocia al mensaje SEI del período de almacenamiento intermedio que inicializa la CPB puede denominarse como la unidad de acceso 0. El valor de n puede aumentarse en 1 para cada unidad de acceso posterior en el orden de decodificación.

30 Cada unidad de decodificación puede denominarse respectivamente como una unidad de decodificación m , donde el número m identifica la unidad de decodificación particular. La primera unidad de decodificación en orden de decodificación en la unidad de acceso 0 puede denominarse como la unidad de decodificación 0. El valor de m puede aumentarse en 1 para cada unidad de decodificación posterior en orden de decodificación.

35 En este ejemplo, si la variable `SubPicCpbFlag` es igual a 0, la variable `InitCpbRemovalDelay[SchedSelIdx]` puede ajustarse en `initial_cpb_removal_delay[SchedSelIdx]` del mensaje SEI del período de almacenamiento intermedio asociado, y `InitCpbRemovalDelayOffset[SchedSelIdx]` puede ajustarse en `initial_cpb_removal_delay_offset[SchedSelIdx]` del mensaje SEI del período de almacenamiento intermedio asociado. De otro modo, la variable `InitCpbRemovalDelay[SchedSelIdx]` puede ajustarse en `initial_du_cpb_removal_delay[SchedSelIdx]` del mensaje SEI del período de almacenamiento intermedio asociado, y `InitCpbRemovalDelayOffset[SchedSelIdx]` puede ajustarse en `initial_du_cpb_removal_delay_offset[SchedSelIdx]` del mensaje SEI del período de almacenamiento asociado.

45 El momento en el que el primer bit de la unidad de decodificación n comienza a entrar en la CPB puede denominarse como el tiempo de llegada inicial $t_{ai}(m)$. La hora de llegada inicial de las unidades de decodificación puede obtenerse como se indica a continuación:

si la unidad de decodificación es la unidad de decodificación 0, $t_{ai}(0) = 0$;
de otro modo (la unidad de decodificación en la unidad de decodificación m con $m > 0$), puede aplicarse lo siguiente:

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

$$t_{ai}(m) = t_{af}(m - 1) \quad (\text{C-2})$$

55 de otro modo (`cbr_flag[SchedSelIdx]` es igual a 0), el tiempo de llegada inicial de la unidad de decodificación m puede obtenerse por

$$t_{ai}(m) = \text{Max}(t_{af}(m - 1), t_{ai,m\acute{a}stemprano}(m)) \quad (\text{C-3})$$

60 donde $t_{ai,m\acute{a}stemprano}(m)$ puede obtenerse como se indica a continuación:

si la unidad de decodificación n no es la primera unidad de decodificación de un período de almacenamiento en memoria intermedia posterior, $t_{ai,m\acute{a}stemprano}(m)$ puede obtenerse como:

$$65 \quad t_{ai,m\acute{a}stemprano}(m) = t_{r,n}(m) - (\text{InitCpbRemovalDelay[SchedSelIdx]} + \text{InitCpbRemovalDelayOffset[SchedSelIdx]})$$

÷ 90000 (C-4)

siendo $t_{r,n}(m)$ el tiempo de eliminación nominal de la unidad de decodificación m de la CPB;

5 de otro modo (la unidad de decodificación n no es la primera unidad de decodificación de un período de almacenamiento en memoria intermedia posterior), $t_{ai,mástemprano}(m)$ puede obtenerse como

$$t_{ai,mástemprano}(m) = t_{r,n}(m) - (\text{InitCpbRemovalDelay}[\text{SchedSelIdx}] \div 90000) \quad (\text{C-5})$$

10 El tiempo de llegada final t_{af} para la unidad de decodificación m puede obtenerse por

$$t_{af}(m) = t_{ai}(m) + b(m) \div \text{BitRate}[\text{SchedSelIdx}] \quad (\text{C-6})$$

15 donde $b(m)$ es el tamaño en bits de la unidad de decodificación m , contando los bits de las unidades NAL VCL y las unidades NAL de datos de relleno para el punto de conformidad de Tipo I o todos los bits del flujo de bits de Tipo II para el punto de la conformidad de Tipo II.

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

20 si el contenido de los conjuntos de parámetros de secuencia activa para la unidad de acceso que contiene la unidad de decodificación m y la unidad de acceso anterior difieren, el HSS puede seleccionar un valor SchedSelIdx1 de SchedSelIdx de entre los valores de SchedSelIdx proporcionados en el conjunto de parámetros de secuencia activa para la unidad de acceso que contiene la unidad de decodificación m que se traduce en

25 $\text{BitRate}[\text{SchedSelIdx1}]$ o $\text{CpbSize}[\text{SchedSelIdx1}]$ para la unidad de acceso que contiene la unidad de decodificación m . El valor de $\text{BitRate}[\text{SchedSelIdx1}]$ o $\text{CpbSize}[\text{SchedSelIdx1}]$ puede diferir del valor de $\text{BitRate}[\text{SchedSelIdx0}]$ o $\text{CpbSize}[\text{SchedSelIdx0}]$ para el valor de SchedSelIdx0 de SchedSelIdx que estaba en uso para la unidad de acceso anterior;

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

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:

35 la variable $\text{BitRate}[\text{SchedSelIdx}]$ puede entrar en vigor en el momento $t_{ai}(m)$

la variable de $\text{CpbSize}[\text{SchedSelIdx}]$ puede entrar en vigor como se indica a continuación:

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

45 de otro modo, el nuevo valor de $\text{CpbSize}[\text{SchedSelIdx}]$ puede entrar en vigor en el momento de la eliminación de la CPB de la última unidad de decodificación de la unidad de acceso que contiene la unidad de decodificación m .

50 Cuando la variable SubPicCpbFlag es igual a 1, el tiempo de llegada de CPB inicial de la unidad de acceso n $t_{ai}(n)$ puede ajustarse en el tiempo de llegada de CPB inicial de la primera unidad de decodificación en la unidad de acceso n , y el tiempo de llegada de CPB final de la unidad de acceso n $t_{af}(n)$ puede ajustarse en el tiempo de llegada de CPB final de la última unidad de decodificación en la unidad de acceso n . Cuando SubPicCpbFlag es igual a 0, cada unidad de decodificación puede ser una unidad de acceso, de tal forma que los tiempos de llegada de CPB inicial y final de la unidad de acceso n pueden ser los tiempos de llegada de CPB inicial y final de la unidad de decodificación m .

55 El siguiente análisis proporciona un ejemplo para la temporización de la eliminación de la unidad de decodificación y la decodificación de una unidad de decodificación. Si SubPicCpbFlag es igual a 0, la variable $\text{CpbRemovalDelay}(m)$ se puede establecer en el valor de cpbremovaldelay especificado en el mensaje SEI de temporización de imagen asociado con la unidad de acceso que es la unidad de decodificación m . De otro modo, $\text{CpbRemovalDelay}(m)$ se puede establecer en el valor de $\text{du_cpb_removal_delay}[i]$ para la unidad de decodificación m especificada en el mensaje SEI de temporización de imagen asociado a la unidad de acceso que contiene unidad de decodificación m .

60 Cuando una unidad de decodificación m es la unidad de decodificación con m igual a 0 (la primera unidad de decodificación de la unidad de acceso que inicializa el HRD), el tiempo de eliminación nominal de la unidad de decodificación de la CPB puede especificarse por:

$$t_{r,n}(0) = \text{InitCpbRemovalDelay}[\text{SchedSelIdx}] \div 90000 \quad (\text{C-7})$$

5 Cuando una unidad de decodificación m es la primera unidad de decodificación de la primera unidad de acceso de un período de almacenamiento en memoria intermedia que no inicializa el HRD, el tiempo de eliminación nominal de la unidad de decodificación de la CPB puede especificarse por:

$$t_{r,n}(m) = t_{r,n}(m_b) + t_c * \text{CpbRemovalDelay}(m) \quad (\text{C-8})$$

10 donde $t_{r,n}(m_b)$ es el tiempo de eliminación nominal de la primera unidad de decodificación del período de almacenamiento en memoria intermedia anterior.

15 Cuando una unidad de decodificación m es la primera unidad de decodificación de un período de almacenamiento en memoria intermedia, m_b puede ajustarse igual a m en el momento de eliminación $t_{r,n}(m)$ de la unidad de decodificación m .

El tiempo de eliminación nominal $t_{r,n}(m)$ de una unidad de decodificación m que no es la primera unidad de decodificación de un período de almacenamiento en memoria intermedia puede darse por:

$$t_{r,n}(m) = t_{r,n}(m_b) + t_c * \text{CpbRemovalDelay}(m) \quad (\text{C-9})$$

20 donde $t_{r,n}(m_b)$ es el tiempo de eliminación nominal de la primera unidad de decodificación del período de almacenamiento en memoria intermedia actual.

La hora de eliminación de la unidad de decodificación m puede especificarse como se indica a continuación:

25 Si `low_delay_hrd_flag` es igual a 0 o $t_{r,n}(m) > \text{taf}(m)$, el tiempo de eliminación de la unidad de decodificación m puede especificarse por:

$$t_r(m) = t_{r,n}(m) \quad (\text{C-10})$$

30 De otro modo, (`low_delay_hrd_flag` es igual a 1 y $t_{r,n}(m) < \text{taf}(m)$), y el tiempo de eliminación de la unidad de decodificación m puede especificarse por:

$$t_r(m) = t_{r,n}(m) + t_c * \text{Ceil}((\text{taf}(m) - t_{r,n}(m)) \div t_c) \quad (\text{C-11})$$

35 El último caso, este ejemplo, indica que el tamaño de la unidad de decodificación m , $b(m)$, es tan grande que se impide la eliminación en el tiempo de eliminación nominal.

40 Cuando `SubPicCpbFlag` es igual a 1, el tiempo de eliminación de CPB nominal de la unidad de acceso n , $t_{r,n}(n)$, puede ajustarse en el tiempo de eliminación de CPB normal de la última unidad de decodificación en la unidad de acceso n ; el tiempo de eliminación de CPB de la unidad de acceso n , $t_r(n)$, puede ajustarse en el tiempo de eliminación de CPB de la última unidad de decodificación en la unidad de acceso n . Cuando `SubPicCpbFlag` es igual a 0, cada unidad de decodificación m es una unidad de acceso n , en este ejemplo y, por lo tanto, el tiempo de eliminación de CPB nominal y el tiempo de eliminación de CPB de la unidad de acceso n son el tiempo de eliminación de CPB nominal y el tiempo de eliminación de CPB de la unidad de decodificación m . En el tiempo de eliminación de CPB de la unidad de decodificación m , en algunos ejemplos, la unidad de decodificación puede decodificarse instantáneamente.

50 En este ejemplo, la memoria intermedia de imagen decodificada (DPB) puede operar como se indica a continuación. La memoria intermedia de imágenes decodificadas puede contener una o más memorias intermedias de almacenamiento de imágenes. Cada una de las memorias intermedias de almacenamiento de imágenes puede contener una imagen decodificada que se marca como "usada para referencia" o se guarda para una salida futura. Antes del inicio, la DPB puede estar vacía (la variable de plenitud de la DPB puede ajustarse a cero). Las siguientes etapas de este ejemplo pueden tener lugar en la secuencia enumerada.

55 En primer lugar, las imágenes pueden eliminarse de la DPB. La eliminación de imágenes de la DPB antes de la decodificación de la imagen actual (pero después de analizar el encabezado de fragmento del primer fragmento de la imagen actual) puede tener lugar instantáneamente en el momento de la eliminación de la CPB de la primera unidad de decodificación de la unidad de acceso n (que contiene la imagen actual) y puede avanzar como se indica a continuación. Puede invocarse el proceso de decodificación para la imagen de referencia ajustada como se especifica en la subcláusula 8.3.2 de HEVC WD6. Si la imagen actual es una imagen imagen IDR, puede aplicarse lo siguiente: cuando la imagen IDR no es la primera imagen IDR decodificada y el valor de `pic_width_in_luma_samples` o `pic_height_in_luma_samples` o `max_dec_pic_buffering` obtenido del conjunto de parámetros de secuencia activos es diferente del valor de `pic_width_in_luma_samples` o `pic_height_in_luma_samples` o `max_dec_pic_buffering` obtenido del conjunto de parámetros de secuencia que estuvo activo para la imagen anterior, respectivamente, `no_output_of_prior_pics_flag` (es decir, una salida nula del

indicador de imágenes anterior) puede deducirse como igual a 1 por el HRD, o ajustarse a 1 para su propio procesamiento por el HRD, independientemente del valor real de no_output_of_prior_pics_flag. Las implementaciones del decodificador pueden intentar manipular los cambios en la imagen o del tamaño de DPB más fácilmente que el HRD con respecto a cambios en pic_width_in_luma_samples o pic_height_in_luma_samples.

5 Cuando no_output_of_prior_pics_flag es igual a 1 o se deduce que es igual a 1, todas las memorias intermedias de almacenamiento de imagen en la DPB pueden estar vacías sin salida de las imágenes que contienen, y la plenitud de DPB se ajusta a 0. (El procesamiento adicional que se puede realizar si no_output_of_prior_pics_flag tiene un valor que no es igual a 1 se describe adicionalmente más adelante). Todas las imágenes k en la DPB, para las que
10 las siguientes condiciones son verdaderas, se eliminan de la DPB:

la imagen k se marca como "no usada para referencia";

15 la imagen k tiene PicOutputFlag igual a 0 o su tiempo de salida DPB es menor de o igual al tiempo de eliminación de la CPB de la primera unidad de decodificación (representada como unidad de decodificación m) de la imagen actual n; es decir $t_{o,dpb}(k) \leq t_r(m)$.

20 Cuando una imagen se elimina de la DPB, la plenitud de la DPB puede reducirse en uno. Para la salida de imagen, lo siguiente puede suceder instantáneamente en el tiempo de eliminación de la CPB de la unidad de acceso n, $t_r(n)$:

La imagen n puede considerarse como decodificada después de que se decodifique la última unidad de decodificación de la imagen.

25 Cuando la imagen n tiene PicOutputFlag igual a 1, su tiempo de salida de DPB $t_{o,dpb}(n)$ puede obtenerse por:

$$t_{o,dpb}(n) = t_r(n) + t_c * dpb_output_delay(n) \quad (C-12)$$

30 donde dpb_output_delay(n) es el valor de dpb_output_delay especificado en el mensaje de SEI de temporización de imagen asociado a la unidad de acceso n.

La salida de la imagen actual puede especificarse como se indica a continuación.

si PicOutputFlag es igual a 1 y $t_{o,dpb}(n) = t_r(n)$, la imagen actual puede transmitirse;

35 por el contrario, si PicOutputFlag es igual a 0, la imagen actual puede no transmitirse, sino que puede almacenarse en la DPB;

40 de lo contrario (PicOutputFlag es igual a 1 y $t_{o,dpb}(n) > t_r(n)$), la imagen actual puede transmitirse posteriormente y puede almacenarse en la DPB (como se especifica en la subcláusula C.3.3 de HEVC WD6 como se modifica por esta divulgación), y puede emitirse en el momento $t_{o,dpb}(n)$, a menos que se indique que no se emite por la decodificación o inferencia de no_output_of_prior_pics_flag igual 1 en un momento que antecede a $t_{o,dpb}(n)$. En otras palabras, la imagen actual puede almacenarse en la DPB y puede transmitirse posteriormente, por ejemplo, en el momento $t_{o,dpb}(n)$, si la no salida del indicador de imágenes anterior no es igual a 1.

45 Al transmitirse, la imagen puede recortarse, usando el rectángulo de recorte especificado en el conjunto de parámetros de secuencia activa.

50 Cuando la imagen n es una imagen que se emite y no es la última imagen del flujo de bits que se emite, el valor de un intervalo de tiempo de salida de DPB $\Delta t_{o,dpb}(n)$ puede definirse como se indica a continuación:

$$\Delta t_{o,dpb}(n) = t_{o,dpb}(n_n) - t_{o,dpb}(n) \quad (C-13)$$

55 donde n_n puede indicar la imagen que sigue después de la imagen n en el orden de salida y tiene PicOutputFlag igual a 1, de tal forma que el intervalo del tiempo de salida de DPB $\Delta t_{o,dpb}(n)$ puede definirse como la diferencia entre el tiempo de salida de DPB de una imagen posterior después de la imagen n en orden de salida y el posterior tiempo de salida de DPB de la imagen n.

60 Para el marcado y almacenamiento de la imagen actual decodificada, puede implementarse lo siguiente de forma instantánea en el momento de la eliminación de CPB de la unidad de acceso n, $t_r(n)$: la imagen decodificada actual puede almacenarse en la DPB en una memoria intermedia de almacenamiento de imagen vacía, y la plenitud de DPB puede aumentarse en uno; si la imagen actual es una imagen de referencia, puede marcarse como "utilizada como referencia", de lo contrario, se puede marcar como "no utilizada como referencia".

65 Para la operación de orden de salida de la DPB, la memoria intermedia de imagen decodificada puede contener una o más memorias intermedias de almacenamiento de imagen. Cada una de las memorias intermedias de almacenamiento de imágenes puede contener una imagen decodificada que se marca como "usada para referencia"

o se guarda para una salida futura. En el inicio del HRD, la DPB puede estar vacía. Las siguientes etapas pueden tener lugar en el orden enumerado.

5 Las imágenes pueden eliminarse de la DPB como se indica a continuación. La eliminación de imágenes de la DPB antes de la decodificación de la imagen actual (pero después de analizar el encabezado del fragmento del primer fragmento de la imagen actual) puede implementarse instantáneamente cuando la primera unidad de decodificación de la unidad de acceso que contiene la imagen actual se elimina de la CPB y puede avanzar como se indica a continuación.

10 El proceso de decodificación para la imagen de referencia fijado como se especifica en la subcláusula 8.3.4.3 de HEVC WD6, como se modifica de acuerdo con esta divulgación, puede invocarse (como se ha descrito parcialmente anteriormente, y como se describe adicionalmente a continuación).

15 Si la imagen actual es una imagen IDR, puede aplicarse lo siguiente:

cuando la imagen IDR no es la primera imagen IDR decodificada y el valor de `pic_width_in_luma_samples` o `pic_height_in_luma_samples` o `max_dec_pic_buffering` obtenido del conjunto de parámetros de secuencia activos es diferente del valor de `pic_width_in_luma_samples` o `pic_height_in_luma_samples` o `max_dec_pic_buffering` obtenido del conjunto de parámetros de secuencia que estuvo activo para la imagen anterior, respectivamente, `no_output_of_prior_pics_flag` puede ajustarse a o inferirse a ser igual a 1 por el HRD, independientemente del valor real de `no_output_of_prior_pics_flag`. Las implementaciones de decodificador pueden tratar de manejar los cambios en el valor de `pic_width_in_luma_samples` o `pic_height_in_luma_samples` o `max_dec_pic_buffering` más fácilmente que el HRD;

25 cuando `no_output_of_prior_pics_flag` es igual a 1 o se deduce que es igual a 1, todas las memorias intermedias de almacenamiento de imagen en la DPB pueden estar vacías están vacías sin salida de las imágenes que contienen;

30 de otro modo, las memorias intermedias de almacenamiento de imágenes que contienen una imagen que se marca como "no necesaria para la salida" y "no usada para referencia" pueden estar vacías (sin salida).

35 Cuando cualquiera de las siguientes condiciones son verdaderas, el proceso de "abultamiento" especificado en la subcláusula C.5.2.1 de HEVC WD6 como se modifica por esta divulgación puede invocarse repetidamente hasta que hay una memoria intermedia de almacenamiento de imágenes vacía para almacenar la imagen decodificada actual:

el número de imágenes en la DPB que están marcadas como "necesaria para la salida" es mayor que varias de las imágenes reordenadas en una capa temporal actual, es decir, `num_reorder_pics [temporal_id]`; o,

40 el número de imágenes en la DPB con el número de imágenes en la DPB con valores de identificador de capa temporal, `temporal_id`, inferior o igual a un valor identificador de capa temporal, `temporal_id`, de la imagen actual es igual a un valor máximo de almacenamiento intermedio de imagen de la capa temporal actual más uno, es decir, `max_dec_pic_buffering[temporal_id] + 1`; o,

45 cuando la imagen actual es una imagen IDR para la que la no salida del indicador de imágenes anterior, `no_output_of_prior_pics_flag`, tiene un valor no igual a 1 y no inferido para ser igual a 1.

50 Pueden realizarse las siguientes etapas: Las memorias intermedias de almacenamiento de imagen que contienen una imagen que se marca como "no necesaria para la salida" y "no marcada para referencia" pueden estar vacías (sin salida), y todas las memorias intermedias de almacenamiento de imágenes no vacías en la DPB pueden vaciarse invocando repetidamente el proceso de "abultamiento" especificado a continuación.

Por lo tanto, un proceso de "abultamiento" puede invocarse en cualquiera de los siguientes casos:

55 la imagen actual es una imagen IDR y `no_output_of_prior_pics_flag` no es igual a 1 y no se ajusta a ni se infiere que es igual a 1, como se especifica en la subcláusula C.5.2 de HEVC WD6 como se modifica por esta divulgación; o,

60 que están marcadas como "necesaria para la salida" es mayor que varias de las imágenes reordenadas en una capa temporal actual, es decir, `num_reorder_pics [temporal_id]`, como se especifica en la subcláusula C.5.2 de HEVC WD6 como se modifica por esta divulgación; o,

65 el número de cuadros en la DPB con `temporal_id` inferior o igual a un valor identificador de capa temporal, `temporal_id`, de la imagen actual es igual a un valor máximo de almacenamiento intermedio de imagen de la capa temporal actual más uno, es decir, `max_dec_pic_buffering[temporal-id] + 1`, como se especifica en la subcláusula C.5.2 de HEVC WD6 como se modifica por esta divulgación.

El proceso de "abultamiento" puede incluir las siguientes etapas ordenadas:

- 5 1. La imagen que está primero para la salida puede seleccionarse como la única que tiene el valor más pequeño de PicOrderCntVal de todas las imágenes en la DPB marcada como "necesaria para la salida".
2. La imagen se recorta, usando el rectángulo de recorte especificado en el conjunto de parámetros de secuencia para la imagen, la imagen recortada puede emitirse, y la imagen puede marcarse como "no necesaria para la salida".
- 10 3. Si la memoria intermedia de almacenamiento de imagen que incluía la imagen que se recortó y se emitió contiene una imagen marcada como "no usada para referencia", la memoria intermedia de almacenamiento de imágenes puede estar vacía.

15 Lo siguiente puede tener lugar instantáneamente para la decodificación, marcado y almacenamiento de imágenes, cuando la última unidad de decodificación de la unidad de acceso n que contiene la imagen actual se elimina de la CPB.

La imagen actual puede considerarse como decodificada después de que se decodifique la última unidad de decodificación de la imagen. La imagen decodificada actual puede almacenarse en una memoria de almacenamiento de imagen vacía en la DPB, y puede aplicarse lo siguiente:

- 20 si la imagen decodificada actual tiene PicOutputFlag igual a 1, puede marcarse como "necesaria para la salida";
- 25 de lo contrario, (la imagen decodificada actual tiene PicOutputFlag igual a 0), puede marcarse como "no necesaria para la salida";

Si la imagen decodificada actual es una imagen de referencia, puede ser marcada como "utilizada como referencia"; de lo contrario (la imagen decodificada actual es una imagen no de referencia), se puede marcar como "no utilizada como referencia".

30 Por lo tanto, un proceso de abultamiento puede incluir: seleccionar una imagen que tiene un menor valor de recuento de orden de imágenes (POC) de imágenes en la DPB y que está marcado según sea necesario para la salida como una imagen seleccionada; recortar la imagen seleccionada como se especifica en un conjunto de parámetros de secuencia activa para la imagen seleccionada, generando de este modo una imagen recortada en base a la imagen seleccionada; transmitir la imagen recortada; y marcar la imagen seleccionada como no sea necesario para la emisión.

40 Pueden usarse elementos de sintaxis para señalar un modo de comportamiento CPB, utilizando la semántica definida a continuación. La sintaxis y de los parámetros VUI y la semántica puede cambiarse para que sea como se muestra en la Tabla 3 a continuación (la semántica de los elementos de sintaxis existentes no se cambian en relación con HEVC WD6, en este ejemplo):

Tabla 3

vui_parameters() {	Descriptor
aspect_ratio_info_present_flag	u(1)
if(aspect_ratio_info_present_flag) {	
aspect_ratio_idc	u(8)
if(aspect_ratio_idc == Extended_SAR) {	
sar_width	u(16)
sar_height	u(16)
}	
}	
overscan_info_present_flag	u(1)
if(overscan_info_present_flag)	
overscan_appropriate_flag	u(1)
video_signal_type_present_flag	u(1)
if(video_signal_type_present_flag) {	
video_format	u(3)
video_full_range_flag	u(1)
colour_description_present_flag	u(1)
if(colour_description_present_flag) {	
colour_primaries	u(8)
transfer_characteristics	u(8)

matrix_coefficients	u(8)
}	
}	
chroma_loc_info_present_flag	u(1)
if(chroma_loc_info_present_flag) {	
chroma_sample_loc_type_top_field	ue(v)
chroma_sample_loc_type_bottom_field	ue(v)
}	
neutral_chroma_indication_flag	u(1)
field_indication_presence_flag	u(1)
timing_info_present_flag	u(1)
if(timing_info_present_flag) {	
num_units_in_tick	u(32)
time_scale	u(32)
fixed_pic_rate_flag	u(1)
}	
nal_hrd_parameters_present_flag	u(1)
if(nal_hrd_parameters_present_flag)	
hrd_parameters()	
vcl_hrd_parameters_present_flag	u(1)
if(vcl_hrd_parameters_present_flag)	
hrd_parameters()	
if(nal_hrd_parameters_present_flag vcl_hrd_parameters_present_flag) {	
sub_pic_cpb_params_present_flag	u(1)
low_delay_hrd_flag	u(1)
}	
bitstream_restriction_flag	u(1)
if(bitstream_restriction_flag) {	
motion_vectors_over_pic_boundaries_flag	u(1)
max_bytes_per_pic_denom	ue(v)
max_bits_per_min_cu_denom	ue(v)
log2_max_mv_length_horizontal	ue(v)
log2_max_mv_length_vertical	ue(v)
}	
}	

En el ejemplo de la Tabla 3, los parámetros VUI incluyen un indicador añadido `sub_pic_cpb_params_present_flag`, en relación con la HEVC convencional. La semántica para este indicador puede definirse como se indica a continuación: `sub_pic_cpb_params_present_flag` que es igual a 1 puede especificar que los parámetros de retardo de eliminación de CPB a nivel de subimagen están presentes y la CPB puede funcionar a nivel de unidad de acceso o a nivel de subimagen. La variable `sub_pic_cpb_flag` que es igual a 0 puede especificar que los parámetros de retardo de eliminación de CPB a nivel de subimagen no están presentes y la CPB tiene que funcionar a nivel de unidad de acceso. Cuando `sub_pic_cpb_params_present_flag` no está presente, su valor puede ajustarse a o inferirse para ser igual a 0.

También pueden usarse elementos de sintaxis para indicar los tiempos de eliminación de CPB de unidades de decodificación, usando la semántica que se describe a continuación. En este ejemplo, los elementos de sintaxis pueden indicarse en un mensaje SEI del período de almacenamiento intermedio, por ejemplo, de acuerdo con el ejemplo de la Tabla 4:

Tabla 4

	Descriptor
<code>buffering_period(payloadSize) {</code>	
seq_parameter_set_id	ue(v)
if(IndicadorNalHrdBpPresente) {	
for(SchedSelIdx = 0; SchedSelIdx <= cpb_cnt_minus1; SchedSelIdx++) {	
initial_cpb_removal_delay [SchedSelIdx]	u(v)
initial_cpb_removal_delay_offset [SchedSelIdx]	u(v)
if(sub_pic_cpb_flag) {	
initial_du_cpb_removal_delay [SchedSelIdx]	u(v)
initial_du_cpb_removal_delay_offset [SchedSelIdx]	u(v)
}	
}	
}	

} }	
if(VclHrdBpPresentFlag) {	
for(SchedSelIdx = 0; SchedSelIdx <= cpb_cnt_minus1; SchedSelIdx++) {	
initial_cpb_removal_delay [SchedSelIdx]	u(v)
initial_cpb_removal_delay_offset [SchedSelIdx]	u(v)
if(sub_pic_cpb_flag) {	
initial_du_cpb_removal_delay [SchedSelIdx]	u(v)
initial_du_cpb_removal_delay_offset [SchedSelIdx]	u(v)
}	
}	
}	
}	

En el ejemplo de la Tabla 4, el mensaje SEI del período de memoria intermedia incluye una condicional añadida con respecto a la HEVC convencional, que incluye además, cuando sub_pic_cpb_flag es cierto, que se añaden dos elementos de sintaxis, **initial_du_cpb_removal_delay** [SchedSelIdx] e **initial_du_cpb_removal_delay_offset** [SchedSelIdx]. Esta condicional y elementos de sintaxis añadidos pueden añadirse dentro de una o ambas de las condicionales para cuando NalHrdBpPresentFlag es cierta y/o cuando VclHardBpPresentFlag es cierta.

La Tabla 5 proporciona un ejemplo alternativo en el que se define un mensaje SEI diferente para señalar el retraso inicial de eliminación de CPB y la compensación de eliminación de CPB inicial para la operación de CPB a nivel de subimagen:

Tabla 5

	Descriptor
du_buffering_period(payloadSize) {	
id_de_conjunto_de_parámetros_secuenciales	ue(v)
if(NalHrdBpPresentFlag) {	
for(SchedSelIdx = 0; SchedSelIdx <= cpb_cnt_minus1; SchedSelIdx++) {	
initial_du_cpb_removal_delay [SchedSelIdx]	u(v)
initial_du_cpb_removal_delay_offset [SchedSelIdx]	u(v)
}	
}	
if(VclHrdBpPresentFlag) {	
for(SchedSelIdx = 0; SchedSelIdx <= cpb_cnt_minus1; SchedSelIdx++) {	
initial_du_cpb_removal_delay [SchedSelIdx]	u(v)
initial_du_cpb_removal_delay_offset [SchedSelIdx]	u(v)
}	
}	
}	

En el ejemplo de la Tabla 4 anterior, cuando NalHrdBpPresentFlag o VclHrdBpPresentFlag son iguales a 1, un mensaje SEI del período de almacenamiento intermedio puede asociarse a cualquier unidad de acceso en el flujo de bits, y un mensaje SEI del período de almacenamiento intermedio puede estar asociado a cada unidad de acceso IDR, con cada unidad de acceso CRA, y con cada unidad de acceso asociado a un mensaje SEI del punto de recuperación. Para algunas aplicaciones, la presencia frecuente de un mensaje SEI del periodo de almacenamiento intermedio puede ser deseable. En algunos ejemplos, se puede especificar un período de memoria intermedia como el conjunto de unidades de acceso entre las dos casos del mensaje SEI del período de almacenamiento intermedio en orden de decodificación.

En los ejemplos de las Tablas 4 y 5 anteriores, la variable seq_parameter_set_id variable puede especificar el conjunto de parámetros de secuencia que contiene los atributos HRD de secuencia. El valor de seq_parameter_set_id puede ser igual al valor de seq_parameter_set_id en el conjunto de parámetros de imagen al que se hace referencia por la imagen codificada primaria asociada al mensaje SEI del periodo de almacenamiento intermedio. En algunos ejemplos, el valor de seq_parameter_set_id puede estar en el intervalo de 0 a 31, inclusive.

En el ejemplo de la Tabla 4 anterior, **initial_cpb_removal_delay** [SchedSelIdx] puede especificar el retardo para la SchedSelIdx-ésima CPB entre la hora 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 intermedio y el momento de la eliminación de la CPB de los datos codificados asociados a la misma unidad de acceso, para el primer período de abultamiento después de la inicialización de HRD. Este 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 en este ejemplo. El elemento de sintaxis **initial_cpb_removal_delay**[SchedSelIdx] puede no ser igual a 0, en este ejemplo, y puede no

exceder $90000 * (\text{CpbSize}[\text{SchedSelIdx}] \text{BitRate}[\text{SchedSelIdx}])$, el tiempo equivalente del tamaño de la CPB en unidades de reloj de 90 kHz, en este ejemplo.

5 En el ejemplo de la Tabla 4, el elemento de sintaxis `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. El elemento de sintaxis `initial_cpb_removal_delay_offset[SchedSelIdx]` puede ser en unidades de un reloj de 90 kHz en este ejemplo. El elemento de sintaxis `initial_cpb_removal_delay_offset[SchedSelIdx]` puede ser un código de longitud fija cuya longitud en bits se da por `initial_cpb_removal_delay_length_minus1 + 1`. Este elemento de sintaxis puede no usarse por los decodificadores y puede ser necesario sólo para el programador de entrega (HSS) especificado en el Anexo C de HEVC WD6. Durante toda la secuencia del vídeo codificado, la suma de `initial_cpb_removal_delay[SchedSelIdx]` e `initial_cpb_removal_delay_offset[SchedSelIdx]` puede ser constante para cada valor de `SchedSelIdx`.

15 En los ejemplos de las Tablas 4 y 5 anteriores, el elemento de sintaxis `initial_du_cpb_removal_delay[SchedSelIdx]` puede especificar el retarde para la `SchedSelIdx`-ésima CPB entre el momento de llegada a la CPB del primer bit de los datos codificados asociados a la primera unidad de decodificación en la unidad de acceso asociada al mensaje SEI del periodo de almacenamiento intermedio y el momento de la eliminación de la CPB de los datos codificados asociados a la misma unidad de decodificación, para el primer período de almacenamiento intermedio después de la inicialización HRD. Este elemento sintáctico puede tener una longitud en bits dada por `initial_cpb_removal_delay_length_minus1 + 1`. Este elemento de sintaxis puede ser en unidades de un reloj de 90 kHz en este ejemplo. En este ejemplo, el elemento de sintaxis `initial_du_cpb_removal_delay[SchedSelIdx]` puede no ser igual a 0 y puede no exceder $90000 * (\text{CpbSize}[\text{SchedSelIdx}] \div \text{BitRate}[\text{SchedSelIdx}])$, el tiempo equivalente del tamaño de la CPB en unidades de reloj de 90 kHz.

25 En los ejemplos de las Tablas 4 y 5 anteriores, el elemento de sintaxis `initial_du_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 decodificación a la CPB. El elemento de sintaxis `initial_cpb_removal_delay_offset[SchedSelIdx]` puede ser en unidades de un reloj de 90 kHz en este ejemplo. El elemento de sintaxis `initial_du_cpb_removal_delay_offset[SchedSelIdx]` puede ser un código de longitud fija cuya longitud en bits se da por `initial_cpb_removal_delay_length_minus1 + 1`. Este elemento de sintaxis puede no usarse por los decodificadores y puede ser necesario sólo para el programador de entrega (HSS) especificado en el Anexo C de HEVC WD6 en este ejemplo.

35 Durante toda la secuencia del vídeo codificado, la suma de `initial_du_cpb_removal_delay[SchedSelIdx]` e `initial_du_cpb_removal_delay_offset[SchedSelIdx]` puede ser constante para cada valor de `SchedSelIdx`.

40 La Tabla 6 a continuación proporciona una sintaxis de mensaje SEI de temporización de imagen:

Tabla 6

	Descriptor
<code>pic_timing(payloadSize) {</code>	
<code> if(CpbDpbDelaysPresentFlag) {</code>	
<code> cpb_removal_delay</code>	<code>u(v)</code>
<code> dpb_output_delay</code>	<code>u(v)</code>
<code> if(sub_pic_cpb_flag) {</code>	
<code> num_decoding_units_minus1</code>	<code>ue(v)</code>
<code> for(i = 0; i <= num_decoding_units_minus1; i++) {</code>	
<code> num_nalus_in_du_minus1[i]</code>	<code>ue(v)</code>
<code> du_cpb_removal_delay[i]</code>	<code>u(v)</code>
<code> }</code>	
<code> }</code>	
<code> }</code>	
<code>}</code>	

45 En este ejemplo, el mensaje SEI de temporización de imagen incluye una declaración condicional añadido para `sub_pic_cpb_flag`, que cuando es cierto indica un elemento de sintaxis `num_decoding_units_minus1` y un bucle que señala, para cada una de las unidades de decodificación, un `num_nalus_in_du_minus1` correspondiente y un `du_cpb_removal_delay`. Como alternativa, el mapeo de las unidades NAL a cada unidad de decodificación se puede señalar usando otros medios, por ejemplo, incluyendo un ID de unidad de decodificación para cada unidad NAL VCL, por ejemplo, en la cabecera de la unidad NAL, la cabecera de la porción, o un nuevo mensaje SEI. El ID de decodificación para cada unidad NAL no VCL puede ser el mismo que la unidad NAL VCL asociada.

La sintaxis del mensaje SEI de temporización de imagen en el ejemplo de Tabla 6 puede depender del contenido del

conjunto de parámetros de secuencia que es activo para la imagen codificada asociada al mensaje SEI de temporización de imagen. Sin embargo, a menos que el mensaje SEI de temporización de imagen de una unidad de acceso IDR está precedido de un mensaje SEI del período de almacenamiento intermedio dentro de la misma unidad de acceso, la activación del conjunto de parámetros de secuencia asociado (y, para imágenes IDR que no son la primera imagen en el flujo de bits, la determinación de que la imagen codificada es una imagen IDR) puede no producirse hasta la decodificación de la primera unidad NAL de porción codificada de la imagen codificada. Dado que la unidad NAL de porción codificada de la imagen codificada puede seguir al mensaje SEI de temporización de imagen en el orden de la unidad NAL, puede haber casos en los que un decodificador almacena la RBSP que contiene el mensaje SEI de temporización de imagen hasta la determinación de los parámetros del parámetro de secuencia que estarán activos para la imagen codificada, y después realiza el análisis del mensaje SEI de temporización de imagen.

La presencia del mensaje SEI de temporización de imagen, por el ejemplo de la Tabla 6, en el flujo de bits se puede especificar como se indica a continuación.

si `CpbDpbDelaysPresentFlag` es igual a 1, un mensaje SEI de temporización de imagen puede estar en cada unidad de acceso de la secuencia de vídeo codificada;

de otro modo, `CpbDpbDelaysPresentFlag` es igual a 0, ningún mensaje SEI de temporización de imagen ha de estar presente en ninguna unidad de acceso de la secuencia de vídeo codificada.

La variable `cpb_removal_delay` puede especificar cuantos ciclos de reloj (véase la subcláusula E.2.1 de HEVC WD6) esperar después de la eliminación de la CPB de la unidad de acceso asociada al mensaje SEI del periodo de almacenamiento intermedio más reciente en una unidad de acceso anterior antes de eliminarse de la memoria intermedia los datos de la unidad de acceso asociados al mensaje SEI de temporización de imagen. Este valor también puede usarse para calcular un tiempo lo más temprano posible de llegada de datos de la unidad de acceso a la CPB para el HSS, como se especifica en el Anexo C de HEVC WD6. El elemento de sintaxis puede ser un código de longitud fija, cuya longitud en bits se da por `cpb_removal_delay_length_minus1 + 1`. El `cpb_removal_delay` puede ser el resto de un contador de módulo (`cpb_removal_delay_length_minus1 + 1`). El valor de `cpb_removal_delay_length_minus1` que determina la longitud (en bits) del elemento de sintaxis `cpb_removal_delay` puede ser el valor de `cpb_removal_delay_length_minus1` codificado en el conjunto de parámetros de secuencia que está activo para la imagen codificada asociada al mensaje SEI de temporización de imagen. Sin embargo, `cpb_removal_delay` puede especificar un número de ciclos de reloj en relación con el tiempo de eliminación de la unidad de acceso anterior que contiene un mensaje SEI del período de almacenamiento intermedio, que puede ser una unidad de acceso de una secuencia de vídeo codificado diferente.

La variable `dpb_output_delay` puede usarse para computar el tiempo de salida de DPB de la imagen. Esta variable puede especificar el número de ciclos de reloj a esperar después de la eliminación de la última unidad de decodificación de una unidad de acceso de la CPB antes de que la imagen decodificada se transmita desde la DPB (véase la subcláusula C.2. de HEVC WD6). Una imagen puede no eliminarse de la DPB en su tiempo de salida cuando todavía está marcada como "utilizada como referencia a corto plazo" o "utilizada como referencia a largo plazo" en este ejemplo. Sólo se puede especificar una variable `dpb_output_delay` para una imagen decodificada en este ejemplo.

La longitud del elemento de sintaxis `dpb_output_delay` se puede dar en bits por `dpb_output_delay_length_minus1 + 1`. Cuando `max_dec_pic_buffering[max_temporal_layers_minus1]` es igual a 0, `dpb_output_delay` también puede ser igual a 0.

El tiempo de salida obtenido de `dpb_output_delay` de cualquier imagen que se transmite desde un tiempo de salida conforme al decodificador como se especifica en la subcláusula C.2. de HEVC WD6 como se modifica por esta divulgación, puede preceder al momento de salida obtenido de `dpb_output_delay` de todas las imágenes en cualquier secuencia de vídeo codificada posterior en el orden de decodificación.

El orden de salida de imagen establecido por los valores de este elemento de sintaxis puede ser el mismo orden que se establece por los valores de `PicOrderCnt()` como se especifica por la subcláusula C.5 de HEVC WD6.

Para las imágenes que no se transmiten por el proceso de "abultamiento" de la subcláusula C.5 de HEVC WD6 como se modifica por esta divulgación, porque preceden, en orden de decodificación, a una imagen IDR con `no_output_of_prior_pics_flag` igual a 1 o deducirse como igual a 1, los tiempos de salida obtenidos de `dpb_output_delay` pueden estar aumentando con el valor en aumento de `PicOrderCnt()` con respecto a todas las imágenes dentro de la misma secuencia de vídeo codificada.

La variable `num_decoding_units_minus1` más 1 puede especificar el número de unidades de decodificación en la unidad de acceso a la que está asociado mensaje SEI de temporización de imagen. El valor de `num_decoding_units_minus1` puede estar en el intervalo de 0 a X, inclusive, por ejemplo.

La variable `num_nalus_in_du_minus1[i]` plus 1 puede especificar el número de unidades NAL en la *i*-ésima unidad de decodificación de la unidad de acceso a la que está asociado el mensaje SEI de temporización de imagen. El valor de `num_nalus_in_du_minus1 [i]` puede estar en el intervalo de 0 a X, inclusive, por ejemplo.

5 La primera unidad de decodificación de la unidad de acceso puede incluir las primeras unidades NAL consecutivas `num_nalus_in_du_minus1 [0] + 1` en el orden de decodificación en la unidad de acceso. La *i*-ésima (con *i* mayor que 0) unidad de decodificación de la unidad de acceso puede incluir las unidades NAL consecutivas `num_nalus_in_du_minus1 [i] + 1` inmediatamente después de la última unidad NAL en la unidad de decodificación anterior de la unidad de acceso, en orden de decodificación. Puede haber al menos una unidad NAL VCL en cada
10 unidad de decodificación, por ejemplo.

La variable `du_cpb_removal_delay[i]` puede especificar cuantos ciclos de reloj (véase la subcláusula E.2.1 de HEVC WD6) esperar después de la eliminación de la CPB de la primera unidad de decodificación en la unidad de acceso asociada al mensaje SEI del periodo de almacenamiento intermedio más reciente en una unidad de acceso anterior
15 antes de eliminarse de la CPB la *i*-ésima unidad de decodificación en la unidad de acceso asociada al mensaje SEI de temporización de imagen. Este valor también puede usarse para calcular un tiempo lo más temprano posible de llegada de datos de la unidad de decodificación a la CPB para el HSS, como se especifica en el Anexo C de HEVC WD6. El elemento de sintaxis puede ser un código de longitud fija, cuya longitud en bits puede darse por `cpb_removal_delay_length_minus1 + 1`. El `du_cpb_removal_delay[i]` puede ser el resto de un contador de módulo
20 (`cpb_removal_delay_length_minus1 + 1`). El valor de `cpb_removal_delay_length_minus1` que determina la longitud (en bits) del elemento de sintaxis `du_cpb_removal_delay[i]` puede ser el valor de `cpb_removal_delay_length_minus1` codificado en el conjunto de parámetros de secuencia que está activo para la imagen codificada asociada al mensaje SEI de temporización de imagen. Sin embargo, `du_cpb_removal_delay [i]` especifica un número de ciclos de reloj en relación con el tiempo de eliminación de la primera unidad de decodificación en la unidad de acceso
25 anterior que contiene un mensaje SEI del periodo de almacenamiento intermedio, que puede ser una unidad de acceso de una secuencia de vídeo codificado diferente.

La figura 2 es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo 20 que puede implementar técnicas implicadas en el almacenamiento de una o más unidades de decodificación de datos de vídeo en una
30 memoria intermedia de imágenes, para obtener un tiempo de eliminación de memoria intermedia respectivo para las una o más unidades de decodificación, eliminar las unidades de decodificación de la memoria intermedia de imágenes de acuerdo con el tiempo de eliminación de memoria intermedia obtenido para cada una de las unidades de decodificación, y codificar datos de vídeo correspondientes a las unidades de decodificación eliminadas, entre otras funciones, como se describe en esta divulgación. El codificador de vídeo 20 puede realizar la intra e inter-
35 codificación de bloques dentro de tramas de vídeo, incluyendo unidades de codificación (CU), o subCU de CU. La intra-codificación se apoya en la predicción espacial para reducir o eliminar la redundancia espacial en el vídeo dentro de una trama de vídeo dada. La inter-codificación se apoya en la predicción temporal para reducir o eliminar la redundancia temporal en el vídeo dentro de tramas adyacentes de una secuencia de vídeo. El modo intra (modo I) puede referirse a cualquiera de varios modos de compresión con base espacial y los inter-modos tales como la predicción unidireccional (modo P) o predicción bidireccional (modo B), pueden referirse a cualquiera de varios
40 modos de compresión con base temporal. Aunque algunos componentes para la codificación inter-modo se representan en la figura 2, se debe entender que el codificador de vídeo 20 puede incluir además componentes para la codificación intra-modo, tal como la unidad de intra-predicción 46. Los componentes adicionales que también pueden incluirse no se ilustran en la figura 2 con fines de brevedad y claridad.

45 Como se muestra en la figura 2, el codificador de vídeo 20 recibe bloques de vídeo incluyendo un bloque de vídeo actual en una trama de vídeo que va a codificarse. En el ejemplo de la figura 2, el codificador de vídeo 20 incluye una unidad de compensación de movimiento 44, una unidad de estimación de movimiento 42, una memoria de imágenes de referencia 64, un sumador 50, una unidad de transformación 54, una unidad de cuantificación 90, una
50 unidad de codificación por entropía 56, una memoria intermedia 90, y una memoria intermedia de imágenes codificadas 92. Para la reconstrucción de bloques de vídeo, el codificador de vídeo 20 incluye además una unidad de cuantificación inversa 58, una unidad de transformada inversa 60 y un sumador 62.

55 Durante el proceso de codificación, el codificador de vídeo 20 recibe una trama o un fragmento de vídeo que va a codificarse. La trama o el fragmento puede estar dividido en múltiples bloques de vídeo. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 llevan a cabo una codificación de inter-predictiva del bloque de vídeo recibido con respecto a uno o más bloques de uno o más tramas de referencia para proporcionar una compresión temporal. La unidad de intra-predicción 46 también puede llevar a cabo una codificación intra-
60 predictiva del bloque de vídeo recibido con respecto a uno o más bloques vecinos de la misma trama o fragmento que el bloque que va a codificarse para proporcionar compresión espacial.

La unidad de selección de modo 40 puede seleccionar uno de los modos de codificación (intra o inter), por ejemplo en función de los resultados de error, y puede proporcionar el bloque intra-codificado o inter-codificado resultante al sumador 50 para generar datos de bloque residuales y al sumador 62 para reconstruir el bloque codificado para usarse como una trama de referencia.

65

La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden estar sumamente integradas, pero se ilustran por separado con fines conceptuales. La estimación de movimiento es el proceso de generar vectores de movimiento, que estiman el movimiento para los bloques de vídeo. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de un bloque predictivo de una trama de referencia predictiva (u otra unidad codificada) con respecto al bloque actual que está codificándose en la trama actual (u otra unidad codificada). Un bloque predictivo es un bloque que se corresponde estrechamente con el bloque que va a codificarse, en lo que respecta a la diferencia de píxeles, que puede determinarse mediante la suma de una diferencia absoluta (SAD), una suma de diferencia de cuadrados (SSD) u otras métricas de diferencia. Un vector de movimiento también puede indicar el desplazamiento de una partición de un macrobloque. La compensación de movimiento puede implicar extraer o generar el bloque predictivo en función del vector de movimiento determinado por la estimación de movimiento. Como se aprecia, la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden combinarse de manera funcional en algunos ejemplós.

En el caso de la inter-codificación, la unidad de estimación de movimiento 42 calcula un vector de movimiento para el bloque de vídeo de una trama inter-codificada mediante la comparación del bloque de vídeo a los bloques de vídeo de una trama de referencia en la memoria de imágenes de referencia 64. La unidad de compensación de movimiento 44 también puede interpolar píxeles subenteros de la trama de referencia, por ejemplo, una trama I o una trama P. Como un ejemplo, los vectores de movimiento se pueden predecir a partir de dos listas de tramas de referencia: lista 0, que incluye las tramas de referencia que tienen un orden de visualización antes de la codificación de una trama actual, y la lista 1, que incluye las tramas de referencia que tienen un orden de visualización posterior a la trama actual que se está codificando. Por lo tanto, los datos almacenados en la memoria de imágenes de referencia 64 pueden organizarse de acuerdo con estas dos listas de tramas de referencia.

La unidad de estimación de movimiento 42 compara bloques de una o más tramas de referencia de la memoria de imágenes de referencia 64 con un bloque a codificar de una trama actual, por ejemplo, una trama P o una trama B. Cuando las tramas de referencia en la memoria de imágenes de referencia 64 incluyen valores para píxeles de subnúmeros entero, un vector de movimiento calculado por la unidad de estimación de movimiento 42 puede referirse a una ubicación de píxel subentero de una trama de referencia. La unidad de estimación de movimiento 42 y/o la unidad de compensación de movimiento 44 también puede configurarse para calcular valores de posiciones de píxeles subenteros de tramas de referencia almacenados en la memoria de imágenes de referencia 64 si no hay valores de posiciones de píxeles subenteros almacenados en la memoria de imágenes de referencia 64. 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. El bloque de trama de referencia identificado por un vector de movimiento puede denominarse como un bloque predictivo.

La unidad de compensación de movimiento 44 puede calcular los datos de predicción en base al bloque predictivo. El codificador de vídeo 20 forma un bloque de vídeo residual sustrayendo los datos de predicción proporcionados por la unidad de compensación de movimiento 44 del bloque de vídeo original que está codificándose. El sumador 50 representa el componente o los componentes que llevan a cabo esta operación de sustracción. La unidad de transformada 52 aplica una transformada, tal como una transformada discreta del coseno (DCT) o una transformada conceptualmente similar, al bloque residual, generando un bloque de vídeo que comprende valores de coeficientes de transformada residuales. La unidad de transformada 52 puede llevar a cabo otras transformadas, tales como las definidas por el estándar H.264, que son conceptualmente similares a la DCT. Como otros ejemplos, la unidad de transformada 52 puede realizar transformadas de ondícula, las transformadas de números enteros, las transformadas de subbandas, u otros tipos de transformadas. La unidad de transformada 52 aplica la transformada al bloque residual, generando un bloque de coeficientes de transformada residuales. La transformada puede convertir la información residual desde un dominio de valor de píxel a un dominio de transformada, tal como un dominio de frecuencia. La unidad de cuantificación 54 cuantifica los coeficientes de transformación residuales para reducir adicionalmente la velocidad de bits. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos o todos los coeficientes. El grado de cuantificación puede modificarse ajustando un parámetro de cuantificación.

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 la codificación de longitud variable adaptable al contenido (CAVLC), la codificación aritmética binaria adaptable al contexto (CABAC), la codificación de entropía mediante división en intervalos de probabilidad (PIPE) u otra técnica de codificación de entropía. Después de la codificación por entropía por la unidad de codificación de entropía 56, los datos de vídeo codificados pueden almacenarse en memoria intermedia o almacenarse más o menos temporalmente en un memoria intermedia de imagen codificada 92, transmitirse a otro dispositivo, y/o archivar para su posterior transmisión o recuperación. En el caso de una codificación aritmética binaria adaptativa de contexto, el contexto puede basarse en macrobloques vecinos.

En algunos casos, la unidad de codificación de entropía 56 u otra unidad de codificador de vídeo 20 pueden estar configuradas para realizar otras funciones de codificación, además de la codificación de entropía. Por ejemplo, la unidad de codificación de entropía 56 puede estar configurada para determinar los valores de patrón de bloque codificado (CBP) valores para los macrobloques y particiones. Además, en algunos casos, la unidad de codificación

por entropía 56 puede realizar una codificación de longitud de ejecución de los coeficientes en una unidad de codificación mayor (LCU) o una subCU de una LCU. En particular, la unidad de codificación de entropía 56 puede aplicar una exploración en zigzag u otro patrón de exploración para explorar los coeficientes de transformación en una LCU o partición y codificar realizaciones de ceros para una mayor compresión. La unidad de codificación por entropía 56 y/o otros elementos de codificador de vídeo 20 también pueden formar unidades de decodificación de datos de vídeo codificados. Las unidades de decodificación pueden ser subimágenes, tal como una secuencia de bloques de árbol, una o más porciones, una o más ondas, y/o una o más piezas, por ejemplo. La unidad de codificación por entropía 56 y/u otros elementos del codificador de vídeo 20 también pueden añadir datos de relleno para las subimágenes con diferentes tamaños para lograr la alineación de bytes. La unidad de codificación por entropía 56 también puede construir la información de cabecera con los elementos de sintaxis adecuados para la transmisión en el flujo de bits de vídeo codificado. Por ejemplo, la información de cabecera puede incluir datos de señalización que indican si las unidades de decodificación son unidades de acceso o unidades de subacceso. Esto puede incluir la señalización de un valor para un indicador preferido de subimagen memoria intermedia de imágenes codificadas, señalado en los parámetros HRD. La unidad de codificación por entropía 56 y/u otros elementos del codificador de vídeo 20 también pueden añadir elementos de sintaxis tales como mensajes SEI del período de almacenamiento en memoria intermedia, parámetros VUI de señalización, datos de señalización indicativos de los puntos de entrada para las diversas subimágenes, y/o tiempos de eliminación de memoria intermedia para las unidades de decodificación, por ejemplo.

La unidad de cuantificación inversa 58 y la unidad de transformada inversa 60 aplican una cuantificación inversa y una transformación inversa, respectivamente, para reconstruir el bloque residual en el dominio de píxeles, por ejemplo para usarse posteriormente como un bloque de referencia. La unidad de compensación de movimiento 44 puede calcular un bloque de referencia añadiendo el bloque residual a un bloque predictivo de una de las tramas de la memoria de imágenes de referencia 64. La unidad de compensación de movimiento 44 también puede aplicar uno o más filtros de interpolación al bloque residual reconstruido para calcular valores de fracciones de píxel para su uso en la estimación de movimiento. El sumador 62 añade el bloque residual reconstruido al bloque de predicción compensado por movimiento generado por la unidad de compensación de movimiento 44 para generar un bloque de vídeo reconstruido para su almacenamiento en la memoria de imágenes de referencia 64. El bloque de vídeo reconstruido puede usarse por la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 como un bloque de referencia para inter-codificar un bloque en una trama de vídeo posterior.

La memoria de imágenes de referencia 64 puede incluir la memoria intermedia 90. La memoria intermedia 90 puede ser o incluir o incluirse en un dispositivo de almacenamiento de datos, tal como cualquier memoria permanente o volátil capaz de almacenar datos, tal como la memoria síncrona dinámica de acceso aleatorio (SDRAM), memoria de acceso aleatorio dinámica incorporada (eDRAM), o memoria de acceso aleatorio estática (SRAM). La memoria intermedia 90 puede incluir memorias intermedias y/o memorias intermedias de imagen decodificada y puede operar de acuerdo con cualquier combinación de comportamientos de memoria intermedia de imagen codificada ejemplar y/o memoria intermedia de imágenes decodificadas que se describen en esta divulgación. Por ejemplo, el codificador de vídeo 20 puede realizar la gestión de patrones de bloques decodificados (DPB) usando la memoria intermedia 90 y/o la gestión de patrones de bloques codificados (CPB) de la memoria intermedia de imágenes codificadas 92 de acuerdo con técnicas de esta divulgación.

La memoria intermedia de imagen codificada 92 puede ser o incluir o incluirse en un dispositivo de almacenamiento de datos, tal como cualquier memoria permanente o volátil capaz de almacenar datos, tal como la memoria síncrona dinámica de acceso aleatorio (SDRAM), memoria de acceso aleatorio dinámica incorporada (eDRAM), o memoria de acceso aleatorio estática (SRAM). Aunque se muestra como formando parte del codificador de vídeo 20, en algunos ejemplos, la memoria intermedia de imágenes codificadas 92 puede formar parte de un dispositivo, unidad o módulo externo al codificador de vídeo 20. Por ejemplo, la memoria intermedia de imágenes codificadas 92 puede formar parte de una unidad de programador de flujo (o programador de entrega o programador de flujo hipotético (HSS)) externo al codificador de vídeo 20. El codificador de vídeo 20 puede formar unidades de decodificación de datos de vídeo codificados y proporcionar las unidades de decodificación a la unidad de programador de flujo. El codificador de vídeo 20 puede formar las unidades de decodificación con diferentes números de bits o números variables de bloques, en algunos ejemplos. La unidad de programador de flujo puede implementar técnicas de esta descripción para enviar las unidades de decodificación, incluyendo subimágenes, tal como una secuencia de bloques de árbol, una o más porciones, una o más ondas, y/o una más piezas, a un decodificador de vídeo para la decodificación, en los momentos que pueden indicarse por un tiempo de eliminación de memoria intermedia obtenidos (por ejemplo, señalados). El codificador de vídeo 20 pueden formar las unidades de decodificación incluyendo cada una una serie de bloques de codificación dispuestos de forma continua en el orden de decodificación, en algunos ejemplos. La unidad de programador de flujo puede desencapsular adicionalmente unidades de acceso para extraer una o más unidades de capas de abstracción de red (NAL), incluyendo las unidades de decodificación. Asimismo, la unidad de programador de flujo puede desencapsular las unidades NAL para extraer las unidades de decodificación.

El codificador de vídeo 20 puede almacenar unidades de acceso a y eliminar las unidades de acceso de la memoria intermedia de imagen codificada 92 de acuerdo con el comportamiento del decodificador de referencia hipotético (HRD) como se modifica por técnicas de esta divulgación. El codificador de vídeo 20 puede aplicar parámetros HRD, incluyendo retardo de eliminación de CPB inicial, tamaño de la CPB, velocidad de bits, retardo de la salida de DPB

inicial, y el tamaño DPB, así como los tiempos de eliminación de la memoria intermedia para las unidades de decodificación, y valores para los indicadores preferidos de memoria intermedia de imagen codificada de subimagen para indicar si las unidades de decodificación de los datos de vídeo son o no unidades de acceso o subconjuntos de unidades de acceso, por ejemplo. El codificador de vídeo 20 puede formar mensajes SEI en las unidades de acceso
 5 que indican períodos de almacenamiento intermedio y tiempos de eliminación de memoria intermedia para las unidades de decodificación. Por ejemplo, el codificador de vídeo 20 puede proporcionar parámetros de información de usabilidad de vídeo (VUI) con la sintaxis que incluye un indicador de CPB de subimagen, tal como en el ejemplo de la Tabla 1 anterior.

10 Las unidades de decodificación pueden incluir subimágenes de una imagen común, y el codificador de vídeo 20 puede incluir los tiempos de eliminación de memoria intermedia para cada una de las subimágenes de la imagen común en los mensajes SEI para las unidades de acceso. Las diferentes subimágenes se pueden codificar con diferentes cantidades de datos, codificándose algunas subimágenes en diferentes números de bits o bloques, y el
 15 codificador de vídeo 20 puede formar un tiempo de eliminación de memoria intermedia respectivo correspondiente para cada una de las subimágenes de una imagen común. El codificador de vídeo 20 también puede codificar algunas imágenes con subimágenes del mismo tamaño de los datos. Otros componentes también pueden realizar una o más de las funciones atribuidas anteriormente para el codificador de vídeo 20. Por ejemplo, una unidad de encapsulación de un dispositivo de origen (tal como el dispositivo de origen 12 de la figura 1) también pueden formar mensajes SEI que incluyen cualquiera de los parámetros anteriores.

20 Por lo tanto, el codificador de vídeo 20 puede disponer que cada subimagen puede incluir un número de bloques de codificación de una imagen codificada continua en orden de decodificación, y de tal manera que un bloque de codificación puede ser idéntico a un bloque de árbol o un subconjunto de bloques de árbol. El codificador de vídeo
 25 20 puede disponer que la codificación de subimágenes y la asignación de bits a diferentes subimágenes en una imagen puede llevarse a cabo sin necesidad de que cada subimagen (es decir, grupo de bloques de árbol) en una imagen se codifique con la misma cantidad de bits. El codificador de vídeo 20 puede indicar el tiempo de eliminación de la CPB para cada subimagen en el flujo de bits, en lugar de tiempos de eliminación de CPB que se obtienen de acuerdo con los tiempos de eliminación de CPB a nivel de imagen señalados. El codificador de vídeo 20 también
 30 puede incluir más de una subimagen en una porción, y aplicar la alineación de bytes al final de cada subimagen. El codificador de vídeo 20 también puede señalar el punto de entrada de cada subimagen con un valor que indica la alineación de bytes de al menos una de las subimágenes dentro de un conjunto mayor de los datos de vídeo, tales como una porción, una pieza, o una trama, por ejemplo. El codificador de vídeo 20 puede aplicar cualquiera una o más de estas características en diferentes ejemplos de acuerdo con esta divulgación.

35 Las ubicaciones que se muestran para la memoria de imágenes de referencia 64, la memoria intermedia 90, y la memoria intermedia de imágenes codificadas 92 en la figura 2 se muestran con fines ilustrativos. La memoria de imagen de referencia 64, la memoria intermedia 90, y la memoria intermedia de imágenes codificadas 92 pueden situarse en un dispositivo de almacenamiento única o cualquier número de dispositivos de almacenamiento distintos. Los dispositivos de almacenamiento pueden incluir cualquier combinación de medios volátiles y/o no volátiles
 40 legibles por ordenador.

De esta manera, el codificador de vídeo representa un ejemplo de un codificador de vídeo configurado para almacenar una o más unidades de decodificación de datos de vídeo en una memoria intermedia de imágenes, obtener un tiempo de eliminación de memoria intermedia respectivo para las una o más unidades de decodificación,
 45 eliminar las unidades de decodificación de la memoria intermedia de imágenes de acuerdo con el tiempo de eliminación de memoria intermedia obtenido para cada una de las unidades de decodificación, y codificar datos de vídeo correspondientes a las unidades de decodificación eliminadas, entre otras funciones.

50 La figura 3 es un diagrama de bloques que ilustra un ejemplo del decodificador de vídeo 30, que decodifica una secuencia de vídeo codificada. En el ejemplo de la figura 3, el decodificador de vídeo 30 incluye una unidad de decodificación de entropía 70, una unidad de compensación de movimiento 72, una unidad de intra-predicción 74, una unidad de cuantificación inversa 78, una unidad de transformada inversa 80, un sumador 80, una memoria intermedia de imágenes codificadas 94 y una memoria intermedia 96. En algunos ejemplos, el decodificador de vídeo 30 puede llevar a cabo una pasada de decodificación generalmente recíproca a la pasada de codificación
 55 descrita con respecto al codificador de vídeo 20 (figura 2). La unidad de compensación de movimiento 72 puede generar datos de predicción basándose en vectores de movimiento recibidos desde la unidad de decodificación de entropía 70.

60 La unidad de compensación de movimiento 72 puede usar vectores de movimiento recibidos en el flujo de bits para identificar un bloque de predicción en tramas de referencia de la memoria de imágenes de referencia 82. La unidad de intra-predicción 74 puede usar modos de intra-predicción recibidos en el flujo de bits para formar un bloque de predicción a partir de bloques espacialmente adyacentes. La unidad de cuantificación inversa 76 cuantifica de manera inversa, es decir, descuantifica, los coeficientes de bloque cuantificados proporcionados en el flujo de bits y decodificados por la unidad de decodificación de entropía 70. El proceso de cuantificación inversa puede incluir un proceso convencional, por ejemplo, como el definido por la norma de decodificación H.264. El proceso de
 65 cuantificación inversa también puede incluir el uso de un parámetro de cuantificación QPY calculado por un

codificador 20 para cada macrobloque con el fin de determinar el grado de cuantificación y, asimismo, el grado de cuantificación inversa que debería aplicarse.

La unidad de transformada inversa 78 aplica una transformada inversa, por ejemplo una DCT inversa, una transformada inversa de número entero o un proceso de transformada inversa conceptualmente similar, a los coeficientes de transformada con el fin de generar bloques residuales en el dominio de píxeles. La unidad de compensación de movimiento 72 genera bloques compensados por movimiento, posiblemente llevando a cabo una interpolación basada en filtros de interpolación. Los identificadores de los filtros de interpolación que van a usarse para la estimación de movimiento con una precisión de fracción de píxel pueden incluirse en los elementos sintácticos. La unidad de compensación de movimiento 72 puede usar filtros de interpolación como los usados por el codificador de vídeo 20 durante la codificación del bloque de vídeo para calcular valores interpolados para fracciones de píxeles de un bloque de referencia. La unidad de compensación de movimiento 72 puede determinar los filtros de interpolación usados por el codificador de vídeo 20 según la información sintáctica recibida y usar los filtros de interpolación para generar bloques predictivos.

La unidad de compensación de movimiento 72 usa parte de la información sintáctica para determinar tamaños de macrobloques usados para codificar trama(s) de la secuencia de vídeo codificada, información de división que describe cómo cada macrobloque de una trama de la secuencia de vídeo codificada está dividido, modos que indican cómo cada división está codificada, una o más tramas de referencia (y listas de tramas de referencia) para cada macrobloque o división inter-codificado/a y otra información para decodificar la secuencia de vídeo codificada. El sumador 80 suma los bloques residuales a los bloques de predicción correspondientes generados por la unidad de compensación de movimiento 72 o por la unidad de intra-predicción para generar bloques decodificados.

La memoria de imágenes de referencia 82 puede incluir la memoria intermedia 96. La memoria intermedia 96 puede ser o incluir en un dispositivo de almacenamiento de datos, tal como cualquier memoria permanente o volátil capaz de almacenar datos, tal como la memoria síncrona dinámica de acceso aleatorio (SDRAM), memoria de acceso aleatorio dinámica incorporada (eDRAM), o memoria de acceso aleatorio estática (SRAM). La memoria intermedia 96 puede incluir una o más memorias intermedias de imágenes y/o una o más memorias intermedias de imagen decodificada y puede operar de acuerdo con cualquier combinación de comportamientos de memoria intermedia de imagen codificada ejemplar y/o memoria intermedia de imágenes decodificadas que se describen en esta divulgación. Por ejemplo, el decodificador de vídeo 30 puede realizar la gestión DPB usando la memoria intermedia 96 y/o la gestión CPB de la memoria intermedia de imágenes codificadas 94 de acuerdo con las técnicas de esta divulgación.

La memoria intermedia de imágenes codificadas 94 puede implementarse como un dispositivo de almacenamiento de datos, tal como cualquier memoria permanente o volátil capaz de almacenar datos, tal como la memoria síncrona dinámica de acceso aleatorio (SDRAM), memoria de acceso aleatorio dinámica incorporada (eDRAM), o memoria de acceso aleatorio estática (SRAM). La memoria intermedia de imágenes codificadas 94 puede operar de acuerdo con cualquier combinación de comportamientos de memoria intermedia de imágenes codificadas ejemplares desvelados en el presente documento.

Aunque se muestra como formando parte del decodificador de vídeo 30, en algunos ejemplos, la memoria intermedia de imágenes codificadas 94 puede formar parte de un dispositivo, unidad o módulo externo al decodificador de vídeo 30. Por ejemplo, la memoria intermedia de imágenes codificadas 94 puede formar parte de una unidad de programador de flujo externo al decodificador de vídeo 30. La unidad de programador de flujo puede implementar las técnicas de esta descripción para enviar las unidades de decodificación, incluyendo subimágenes, tal como una secuencia de bloques de árbol, una o más porciones, una o más ondas, y/o una más piezas, a un decodificador de vídeo 30 para la decodificación, en los momentos indicados por un tiempo de eliminación de memoria intermedia obtenidos (por ejemplo, señalados). La unidad de programador de flujo puede desencapsular adicionalmente unidades de acceso para extraer una o más unidades de capas de abstracción de red (NAL), incluyendo las unidades de decodificación. Asimismo, la unidad de programador de flujo puede desencapsular las unidades NAL para extraer las unidades de decodificación.

El decodificador de vídeo 30 puede recibir y almacenar unidades de acceso a y eliminar las unidades de acceso de la memoria intermedia de imagen codificada 94 de acuerdo con el comportamiento del decodificador de referencia hipotético (HRD) como se modifica por técnicas de esta divulgación. El decodificador de vídeo 30 puede decodificar y obtener parámetros HRD, incluyendo retardo de eliminación de CPB inicial, tamaño de la CPB, velocidad de bits, retardo de la salida de DPB inicial, y el tamaño DPB, así como los tiempos de eliminación de la memoria intermedia para las unidades de decodificación, y valores para los indicadores preferidos de memoria intermedia de imagen codificada de subimagen que señalizan si las unidades de decodificación de los datos de vídeo son o no unidades de acceso o subconjuntos de unidades de acceso, por ejemplo. El decodificador de vídeo 30 puede decodificar y obtener mensajes SEI en las unidades de acceso que indican períodos de almacenamiento intermedio y tiempos de eliminación de memoria intermedia para las unidades de decodificación. Por ejemplo, el decodificador de vídeo 30 puede decodificar y obtener parámetros de información de usabilidad de vídeo (VUI) con la sintaxis que incluye un indicador de CPB de subimagen, tal como en el ejemplo de la Tabla 1 anterior.

Las unidades de decodificación pueden incluir subimágenes de una imagen común, y el decodificador de vídeo 30 puede decodificar y obtener los tiempos de eliminación de memoria intermedia para cada una de las subimágenes de la imagen común en los mensajes SEI para las unidades de acceso. Las diferentes subimágenes se pueden codificar con diferentes cantidades de datos, codificándose algunas subimágenes en diferentes números de bits o bloques, y el decodificador de vídeo 30 puede decodificar y obtener un tiempo de eliminación de memoria intermedia respectivo correspondiente para cada una de las subimágenes de una imagen común. El decodificador de vídeo 30 también puede decodificar y obtener algunas imágenes con subimágenes del mismo tamaño de los datos.

Por lo tanto, el decodificador de vídeo 30 puede decodificar y obtener subimágenes que pueden incluir un número de bloques de codificación de una imagen codificada continua en orden de decodificación, y de tal manera que un bloque de codificación puede ser idéntico a un bloque de árbol o un subconjunto de bloques de árbol. El decodificador de vídeo 30, en algunos ejemplos, puede decodificar y obtener tiempos de eliminación de la CPB para cada subimagen en el flujo de bits, en lugar de obtener tiempos de eliminación de CPB de acuerdo con los tiempos de eliminación de CPB a nivel de imagen señalados. El decodificador de vídeo 30 también puede decodificar y obtener más de una subimagen en una porción, y puede recibir bytes información de compensación indicativos de los puntos de partida para cada unidad de decodificación, determinar dónde comienza cada unidad de decodificación, y decodificar y obtener información sobre una señal sin datos adicional o señal de relleno que proporciona una alineación de bytes al final de cada subimagen. El decodificador de vídeo 30 también puede obtener el punto de entrada de cada subimagen con un valor que indica la alineación de bytes de al menos una de las subimágenes dentro de un conjunto mayor de los datos de vídeo, tales como una porción, una pieza, o una trama, por ejemplo. El decodificador de vídeo 30 puede aplicar cualquiera una o más de estas características en diferentes ejemplos de acuerdo con esta divulgación.

Las ubicaciones que se muestran para la memoria de imágenes de referencia 82, la memoria intermedia 96, y la memoria intermedia de imágenes codificadas 94 en la figura 3 se muestran con fines ilustrativos. La memoria de imagen de referencia 82, la memoria intermedia 96, y la memoria intermedia de imágenes codificadas 94 pueden situarse en un dispositivo de almacenamiento única o cualquier número de dispositivos de almacenamiento distintos. Los dispositivos de almacenamiento pueden incluir cualquier combinación de medios volátiles y/o no volátiles legibles por ordenador.

De esta manera, el decodificador de vídeo 30 representa un ejemplo de un codificador de vídeo configurado para almacenar una o más unidades de decodificación de datos de vídeo en una memoria intermedia de imágenes, obtener un tiempo de eliminación de memoria intermedia respectivo para las una o más unidades de decodificación, eliminar las unidades de decodificación de la memoria intermedia de imágenes de acuerdo con el tiempo de eliminación de memoria intermedia obtenido para cada una de las unidades de decodificación, y codificar datos de vídeo correspondientes a las unidades de decodificación eliminadas.

La figura 4 es un diagrama de bloques que ilustra un dispositivo de destino ejemplar 100 que puede implementar cualquiera o todas las técnicas de esta divulgación. En este ejemplo, el dispositivo de destino 100 incluye una interfaz de entrada 102, el programador de flujo 104, la memoria intermedia de imágenes codificadas 106, el decodificador de vídeo 108, memoria intermedia de imágenes decodificadas 110, la unidad de representación 112, y la interfaz de salida 114. El dispositivo de destino 100 puede corresponder sustancialmente al dispositivo de origen 14 (figura 1). La interfaz de entrada 102 puede comprender cualquier interfaz de entrada capaz de recibir un flujo de bits codificado de datos de vídeo. Por ejemplo, la interfaz de entrada 102 puede comprender un receptor 26 y/o un módem 28 como en la figura 1, una interfaz de red, tal como una interfaz inalámbrica o por cable, una memoria o una interfaz de memoria, una unidad para leer los datos de un disco, tal como una interfaz de unidad óptica o una interfaz de medios magnéticos, o cualquier otro componente de interfaz.

La interfaz de entrada 102 puede recibir un flujo de bits codificados que incluye datos de vídeo y proporcionar el flujo de bits al programador de flujo 104. De acuerdo con las técnicas de esta divulgación, el programador de flujo 104 extrae unidades de datos de vídeo, tales como unidades de acceso y/o unidades de decodificación, del flujo de bits y almacena las unidades extraídas en la memoria intermedia de imagen codificada 106. De esta manera, el programador de flujo 104 representa un ejemplo de implementación de un HSS como se analiza en los ejemplos anteriores. La memoria intermedia de imágenes codificadas 106 puede adaptarse sustancialmente a la memoria intermedia de imagen codificada 94 (figura 3), excepto que, como se muestra en la figura 4, la memoria intermedia de imágenes codificadas 106 está separada del decodificador de vídeo 108. La memoria intermedia de imágenes codificadas 106 puede estar separada de o integrada como parte del decodificador de vídeo 108 en diferentes ejemplos.

El decodificador de vídeo 108 incluye una memoria intermedia de imágenes decodificadas 110. El decodificador de vídeo 108 puede adaptarse sustancialmente al decodificador de vídeo 30 de las figuras 1 y 3. La memoria intermedia de imágenes decodificadas 110 puede adaptarse sustancialmente a la memoria intermedia 96. Por lo tanto, el decodificador de vídeo 108 puede decodificar las unidades de decodificación de la memoria intermedia de imagen codificada 106 de acuerdo con las técnicas de esta divulgación.

Además, el decodificador de vídeo 108 puede transmitir imágenes decodificadas desde la memoria intermedia de

imagen decodificada 110 de acuerdo con las técnicas de esta divulgación, como se ha analizado anteriormente. El decodificador de vídeo 108 puede transmitir imágenes de salida a la unidad de representación 112. La unidad de representación 112 puede recortar las imágenes como se ha analizado anteriormente, de acuerdo con las técnicas de esta divulgación, después, puede pasar las imágenes recortadas a la interfaz de salida 114. La interfaz de salida 114, a su vez, puede proporcionar las imágenes recortadas a un dispositivo de visualización, que puede adaptarse sustancialmente al dispositivo de visualización 32. El dispositivo de visualización puede formar parte de dispositivo de destino 100, o puede estar acoplado de forma comunicativa al dispositivo de destino 100. Por ejemplo, el dispositivo de visualización puede comprender una pantalla, pantalla táctil, proyector, u otra unidad de visualización integrada con el dispositivo de destino 100, o puede comprender una pantalla independiente, tal como un televisor, monitor, proyector, pantalla táctil u otro dispositivo que está acoplado en comunicación al dispositivo de destino 100. El acoplamiento comunicativo puede comprender un acoplamiento con cable o inalámbrico, tal como mediante un cable coaxial, cable de vídeo compuesto, cable de vídeo componente, un cable de interfaz multimedia de alta definición (HDMI), una emisión de radio-frecuencia, u otro acoplamiento con cable o inalámbrico.

La figura 5 es un diagrama de flujo que ilustra un procedimiento ejemplar que incluye la eliminación de unidades de decodificación de datos de vídeo de una memoria intermedia de imágenes de acuerdo con un tiempo de eliminación de memoria intermedia obtenido, por ejemplo, por el decodificador de vídeo 30 de las figuras 1 o 3 o un decodificador de vídeo 108 de la figura 4 (en conjunto "decodificador de vídeo 30/108"), de acuerdo con técnicas de esta divulgación. El procedimiento ejemplar de la figura 5 puede describirse como realizándose por el decodificador de vídeo 30/108 como un ejemplo, con el entendimiento de que cualquier uno o más aspectos del procedimiento de la figura 5 también pueden realizarse por o implementarse con otros dispositivos o componentes. En el ejemplo de la figura 5, el decodificador de vídeo 30/108 puede almacenar una o más unidades de decodificación de datos de vídeo en una memoria intermedia de imágenes (202). El decodificador de vídeo 30/108 puede obtener un tiempo de eliminación de memoria intermedia respectivo para las una o más unidades de decodificación, donde obtener el tiempo de eliminación de memoria intermedia respectivo comprende recibir un valor señalado respectivo indicativo del tiempo de eliminación de memoria intermedia respectivo para al menos una de las unidades de decodificación (204). El decodificador de vídeo 30/108 puede eliminar las unidades de decodificación de la memoria intermedia de imagen de acuerdo con el tiempo de eliminación de memoria intermedia obtenido para cada una de las unidades de decodificación (206). El decodificador de vídeo 30/108 también puede codificar datos de vídeo correspondientes a las unidades de decodificación eliminadas, donde la codificación de los datos de vídeo comprende la decodificación de la al menos una de las unidades de decodificación (208). El decodificador de vídeo 30/108 y/o otros dispositivos o elementos también pueden realizar funciones diferentes o adicionales en otros ejemplos.

La figura 6 es un diagrama de flujo que ilustra otro procedimiento ejemplar, similar en algunos aspectos al procedimiento de la figura 5, que incluye la eliminación de unidades de decodificación de datos de vídeo de una memoria intermedia de imágenes de acuerdo con un tiempo de eliminación de memoria intermedia obtenido, por ejemplo, por el decodificador de vídeo 30 de las figuras 1 o 3 o un decodificador de vídeo 108 de la figura 4 (en conjunto "decodificador de vídeo 30/108"), de acuerdo con técnicas de esta divulgación. El procedimiento ejemplar de la figura 6 también puede describirse como realizándose por el decodificador de vídeo 30/108 como un ejemplo, con el entendimiento de que cualquier uno o más aspectos del procedimiento de la figura 6 también pueden realizarse por o implementarse con otros dispositivos o componentes. En el ejemplo de la figura 6, el decodificador de vídeo 30/108 puede almacenar una o más unidades de decodificación de datos de vídeo en una memoria intermedia de imagen codificada (CPB) (402), obtener un tiempo de eliminación de memoria intermedia respectivo para las una o más unidades de decodificación (404), eliminar las unidades de decodificación de la CPB de acuerdo con el tiempo de eliminación de memoria intermedia obtenido para cada una de las unidades de decodificación (406), determinar si la CPB funciona a nivel de unidad de acceso o a nivel de subimagen (408), y codificar datos de vídeo correspondientes a las unidades de decodificación eliminadas (410). Si la CPB funciona a nivel de unidad de acceso, la codificación de los datos de vídeo incluye unidades de acceso comprendidas en las unidades de decodificación (412). Si la CPB funciona a nivel de subimagen, la codificación de los datos de vídeo incluye subconjuntos de unidades de acceso comprendidas en las unidades de decodificación (414).

Por ejemplo, si el decodificador de vídeo 30/108 determina que la CPB opera a nivel de unidad de acceso, el decodificador de vídeo 30/108 puede codificar las unidades de acceso de datos de vídeo correspondientes a las unidades de decodificación eliminadas (412). Si el decodificador de vídeo 30/108 determina que la CPB opera a nivel de subimagen, el decodificador de vídeo 30/108 puede codificar subconjuntos de unidades de acceso de datos de vídeo correspondientes a las unidades de decodificación eliminadas (414). Por ejemplo, el decodificador de vídeo 30/108 puede determinar que la una o más unidades de decodificación comprenden unidades de acceso determinando que un indicador preferido de memoria intermedia de imagen codificada de subimagen (por ejemplo, SubPicCpbPreferredFlag) es negativo o tiene un valor de cero o que un indicador presente de parámetros de memoria intermedia de imagen codificada de subimagen (por ejemplo, sub_pic_cpb_params_present_flag) es negativo o tiene un valor de cero. El decodificador de vídeo 30/108 puede determinar que la una o más unidades de decodificación comprenden subconjuntos de unidades de acceso determinando que tanto un indicador preferido de memoria intermedia de imagen codificada de subimagen (por ejemplo, SubPicCpbPreferredFlag) es positivo o tiene un valor de uno, como que un indicador presente de parámetros de memoria intermedia de imagen codificada de subimagen (por ejemplo, sub_pic_cpb_params_present_flag) es positivo o tiene un valor de uno. El decodificador de vídeo 30/108 también puede utilizar un único indicador de memoria intermedia de imagen codificada de subimagen,

SubPicCpbFlag, que puede ajustarse en SubPicCpbPreferredFlag && sub_pic_cpb_params_present_flag, para determinar si ambos indicadores son positivos y que el decodificador de vídeo 30/108 puede codificar subconjuntos de unidades de acceso.

5 La figura 7 es un diagrama de flujo que ilustra otro procedimiento ejemplar de datos de vídeo de procesamiento que incluye la salida de una imagen recortada en un proceso de abultamiento, por ejemplo, por el decodificador de vídeo 30 de las figuras 1 o 3 o un decodificador de vídeo 108 de la figura 4 (en conjunto "decodificador de vídeo 30/108"), de acuerdo con técnicas de esta divulgación. En el ejemplo de la figura 7, el decodificador de vídeo 30/108 puede realizar un proceso de abultamiento si se cumple cualquiera de ciertas condiciones, como se ha descrito
10 anteriormente con referencia a los ejemplos de proceso de abultamiento. En particular, el decodificador de vídeo 30/108 puede realizar un proceso de abultamiento si una imagen actual es una imagen de regeneración de decodificación instantánea (IDR) (302) y una salida nula del indicador de imágenes anterior tiene un valor que no es igual a uno (304), que puede incluir si la salida nula del indicador de imágenes anterior tiene un valor que no se deduce como igual a 1 o ajustarse para que sea igual a 1, por ejemplo, por el HRD. El decodificador de vídeo 30/108
15 también puede realizar un proceso de abultamiento si un número de imágenes en una memoria intermedia de imágenes decodificadas (DPB) que están marcadas como sea necesario para la salida es mayor que un número de imágenes de reordenamiento en una capa temporal actual (306). El decodificador de vídeo 30/108 también puede realizar un proceso de abultamiento si un número de imágenes en la DPB con los valores de identificador de capa temporal inferiores o iguales al valor del identificador de capa temporal de una imagen actual es igual a un valor de almacenamiento intermedio de imagen máximo de la capa temporal actual más uno (308).

Si se cumple cualquiera de las condiciones especificadas (302 y 304, o 306, o 308), el decodificador de vídeo 30/108 puede llevar a cabo un proceso de abultamiento como se indica a continuación. El decodificador de vídeo 30/108 puede seleccionar una imagen que tiene un valor de recuento de orden de imagen menor más pequeño (POC) de
25 imágenes en la DPB y que se marca como sea necesario para la salida como una imagen seleccionada (312). El decodificador de vídeo 30/108 puede recortar la imagen seleccionada como se especifica en un conjunto de parámetros de secuencia activa para la imagen seleccionada, generando de este modo una imagen recortada en base a la imagen seleccionada (314). El decodificador de vídeo 30/108 puede transmitir la imagen recortada (316). El decodificador de vídeo 30/108 puede marcar la imagen seleccionada como no sea necesaria para la transmisión (318).

En uno o más ejemplos, las funciones descritas pueden implementarse en hardware, software, firmware o cualquier combinación de los mismos. Si se implementan en software, las funciones pueden almacenarse en, o transmitirse como una o más instrucciones o código en un medio legible por ordenador, y ejecutarse mediante una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que corresponden a un medio tangible tal como medios de almacenamiento de datos o medios de comunicación que incluyen cualquier medio que facilite la transferencia de un programa informático desde un lugar a otro, por ejemplo, según un protocolo de comunicaciones. De esta manera, los medios legibles por ordenador pueden corresponder, generalmente, a (1) medios de almacenamiento tangibles y legibles por ordenador, que sean no transitorios, o (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible, al que sea pueda acceder por 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.

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 puede denominarse adecuadamente un medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde un sitio web, un servidor u otra fuente remota 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 u otros medios transitorios, sino que, en cambio, se orientan a medios de almacenamiento tangibles no transitorios. Los discos, como se usan en el presente documento, incluyen el disco compacto (CD), el disco de láser, el disco óptico, el disco versátil digital (DVD), el disco flexible y el disco Blu-ray, donde algunos discos normalmente reproducen datos de manera magnética, mientras que otros discos reproducen los datos de manera óptica con láser. Las combinaciones de lo que antecede también deben incluirse dentro del alcance de los medios legibles por ordenador.

Las instrucciones pueden ser ejecutadas por uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados aplicación específica (ASIC), matrices lógicas de campo programable (FPGA) u otros circuitos lógicos integrados o discretos equivalentes. Por consiguiente, el término "procesador", como se usa en el presente documento, puede referirse a cualquier estructura

anterior o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento puede proporcionarse en hardware dedicado y/o módulos de software configurados para la codificación y la decodificación, o incorporarse en un códec combinado. Además, las técnicas podrían implementarse completamente en uno o más circuitos o elementos lógicos.

5

Las técnicas de esta divulgación se pueden implementar en una gran variedad de dispositivos o aparatos, incluyendo un equipo de mano inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). Varios componentes, módulos o unidades se describen en esta divulgación para enfatizar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no requieren necesariamente la realización mediante diferentes unidades de hardware. Más bien, como se ha descrito anteriormente, pueden combinarse diversas unidades en una unidad de hardware de códec o ser proporcionadas por una colección de unidades de hardware interoperativas, incluyendo uno o más procesadores, como se ha descrito anteriormente, junto con el software y/o firmware adecuado.

10

15

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

REIVINDICACIONES

1. Un procedimiento de codificación de datos de vídeo, comprendiendo el procedimiento:
 - 5 almacenar una o más unidades de decodificación de datos de vídeo en una memoria intermedia de imagen codificada (CPB);
 - obtener un tiempo de eliminación de memoria intermedia respectivo para las una o más unidades de decodificación;
 - eliminar las unidades de decodificación de la CPB de acuerdo con el tiempo de eliminación de memoria intermedia obtenido para cada una de las unidades de decodificación;
 - 10 determinar si la CPB funciona a nivel de unidad de acceso o nivel de subimagen; y
 - codificar los datos de vídeo correspondientes a las unidades de decodificación eliminados, en el que, si la CPB funciona a nivel de unidad de acceso, la codificación de los datos de vídeo comprende la codificación de unidades de acceso comprendidas en las unidades de decodificación; y
 - 15 en el que, si la CPB funciona a nivel de subimagen, la codificación de los datos de vídeo comprende subconjuntos de unidades de acceso comprendidas en las unidades de decodificación.

2. El procedimiento de la reivindicación 1, en el que la determinación de si la CPB funciona a un nivel de unidad de acceso o nivel de subimagen comprende:
 - 20 determinar que la CPB opera a nivel de unidad de acceso si un indicador de memoria intermedia de imagen codificada de subimagen preferido tiene un valor de cero o si un indicador de parámetros de memoria intermedia de imagen codificada de subimagen presente tiene un valor de cero; y
 - 25 determinar que la CPB opera a nivel de subimagen si tanto el indicador de memoria intermedia de imagen codificada de subimagen preferido tiene un valor de uno y el indicador de parámetros de memoria intermedia de imagen codificada de subimagen presente tiene un valor de uno.

3. El procedimiento de la reivindicación 2, que comprende además recibir el valor para el indicador de memoria intermedia de imagen codificada de subimagen preferido.

4. El procedimiento de la reivindicación 2, que comprende además establecer un valor de cero para el indicador preferido de memoria intermedia de imagen codificada de subimagen si no se recibe un valor para el indicador preferido de memoria intermedia de imagen codificada de subimagen.

5. El procedimiento de la reivindicación 2, que comprende además recibir el valor para el indicador presente de parámetros de memoria intermedia de imagen codificada de subimagen.

6. El procedimiento de la reivindicación 1, que comprende además:
 - 40 recibir las una o más unidades de decodificación de datos de vídeo en un orden de decodificación continuo,
 - en el que el almacenamiento de las una o más unidades de decodificación de datos de vídeo en la CPB comprende el almacenamiento de las una o más unidades de decodificación de datos de vídeo en el orden de decodificación continuo en la CPB.
 - 45

7. El procedimiento de la reivindicación 1, en el que la obtención del tiempo de eliminación de memoria intermedia respectivo para las una o más unidades de decodificación comprende recibir un valor señalado respectivo indicativo del tiempo de eliminación de memoria intermedia respectivo para al menos una de las unidades de decodificación.

8. El procedimiento de la reivindicación 7, en el que el valor señalado indicativo del tiempo de eliminación de memoria intermedia para la al menos una de las unidades de decodificación comprende un valor señalado indicativo de varias unidades de capa de abstracción de red (NAL) comprendidas por la al menos una de las unidades de decodificación.

9. Un aparato para codificar datos de vídeo, comprendiendo el aparato:
 - 60 medios para almacenar una o más unidades de decodificación de datos de vídeo en una memoria intermedia de imagen codificada (CPB);
 - medios para obtener un tiempo de eliminación de memoria intermedia respectivo para las una o más unidades de decodificación;
 - medios para eliminar las unidades de decodificación de la CPB de acuerdo con el tiempo de eliminación de memoria intermedia obtenido para cada una de las unidades de decodificación;
 - medios para determinar si la CPB funciona a nivel de unidad de acceso o nivel de subimagen; y
 - 65 medios para codificar los datos de vídeo correspondientes a las unidades de decodificación eliminados,

en el que, si la CPB funciona a nivel de unidad de acceso, los medios para codificar los datos de vídeo comprenden medios para codificar unidades de acceso comprendidas en las unidades de decodificación;
y

5 en el que, si la CPB funciona a nivel de subimagen, los medios para codificar los datos de vídeo comprenden medios para codificar subconjuntos de unidades de acceso comprendidas en las unidades de decodificación.

10. El aparato de la reivindicación 9, en el que los medios para determinar si la CPB funciona a nivel de unidad de acceso o nivel de subimagen comprende:

10 medios para determinar que la CPB opera a nivel de unidad de acceso si un indicador de memoria intermedia de imagen codificada de subimagen preferido tiene un valor de cero o si un indicador de parámetros de memoria intermedia de imagen codificada de subimagen presente tiene un valor de cero; y
15 medios para determinar que la CPB opera a nivel de subimagen si tanto el indicador de memoria intermedia de imagen codificada de subimagen preferido tiene un valor de cero o el indicador de parámetros de memoria intermedia de imagen codificada de subimagen presente tiene un valor de uno.

11. El aparato de la reivindicación 10, que comprende adicionalmente:

20 medios para recibir el valor para el indicador de memoria intermedia de imagen codificada de subimagen preferido;
medios para establecer un valor de cero para el indicador preferido de memoria intermedia de imagen codificada de subimagen si no se recibe un valor para el indicador preferido de memoria intermedia de imagen codificada de subimagen; o
25 medios para recibir el valor para el indicador presente de parámetros de memoria intermedia de imagen codificada de subimagen.

12. El aparato de la reivindicación 9, que comprende adicionalmente:

30 medios para recibir las una o más unidades de decodificación de datos de vídeo en un orden de decodificación continuo,
en el que los medios para el almacenamiento de las una o más unidades de decodificación de datos de vídeo en la CPB comprenden medios para el almacenamiento de las una o más unidades de decodificación de datos de vídeo en el orden de decodificación continuo en la CPB.

35 13. El aparato de la reivindicación 9, en el que los medios para obtener el tiempo de eliminación de memoria intermedia respectivo para las una o más unidades de decodificación comprenden medios para recibir un valor señalado respectivo indicativo del tiempo de eliminación de memoria intermedia respectivo para al menos una de las unidades de decodificación.

40 14. El aparato de la reivindicación 13, en el que el valor señalado indicativo del tiempo de eliminación de memoria intermedia para la al menos una de las unidades de decodificación comprende un valor señalado indicativo de varias unidades de capa de abstracción de red (NAL) comprendidas por la al menos una de las unidades de decodificación.

45 15. Un producto de programa informático que comprende un medio legible por ordenador que comprende instrucciones almacenadas en el mismo que, cuando se ejecutan, hacen que un ordenador realice el procedimiento según una cualquiera de la reivindicación 1 a 8.

50

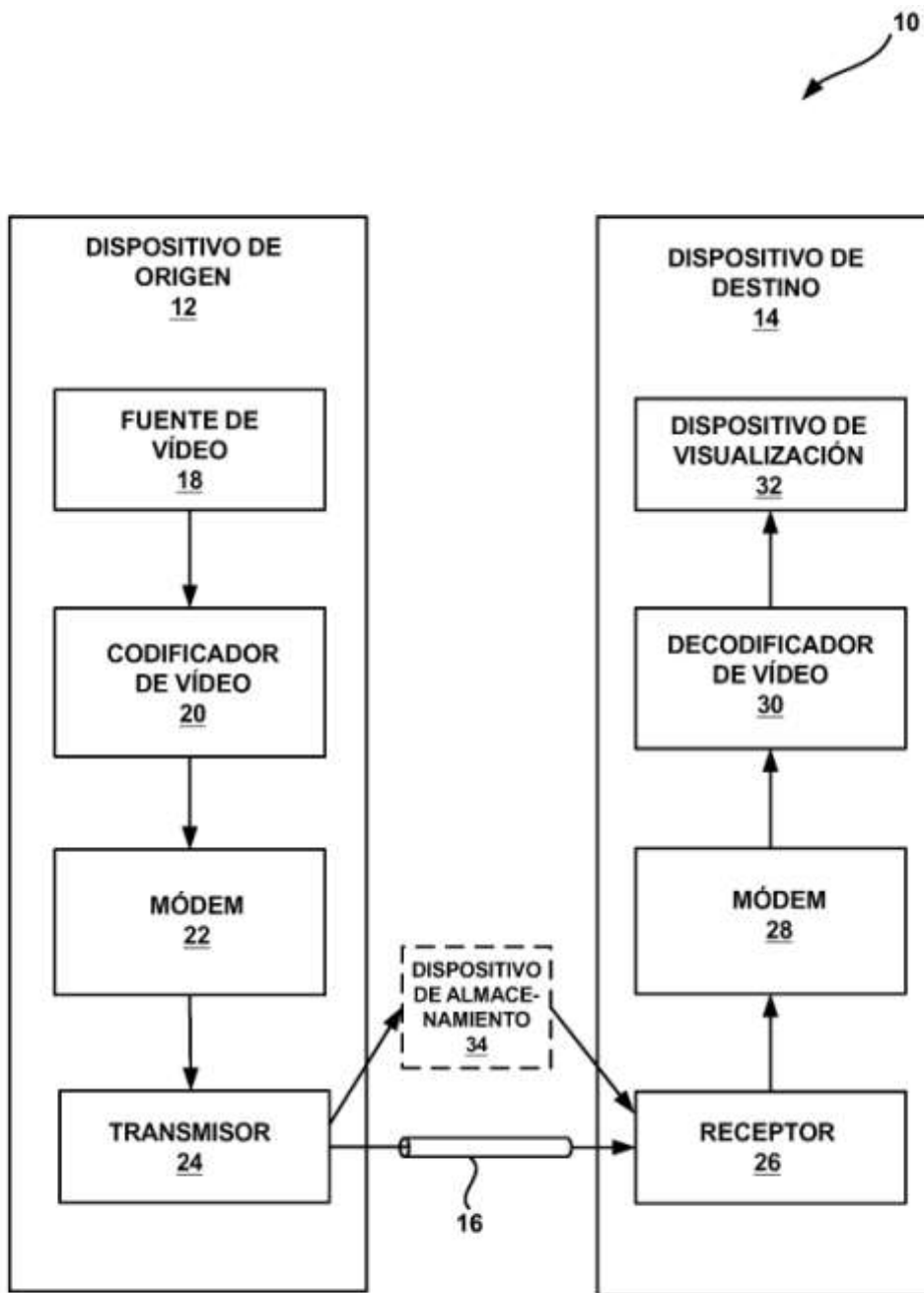


FIG. 1

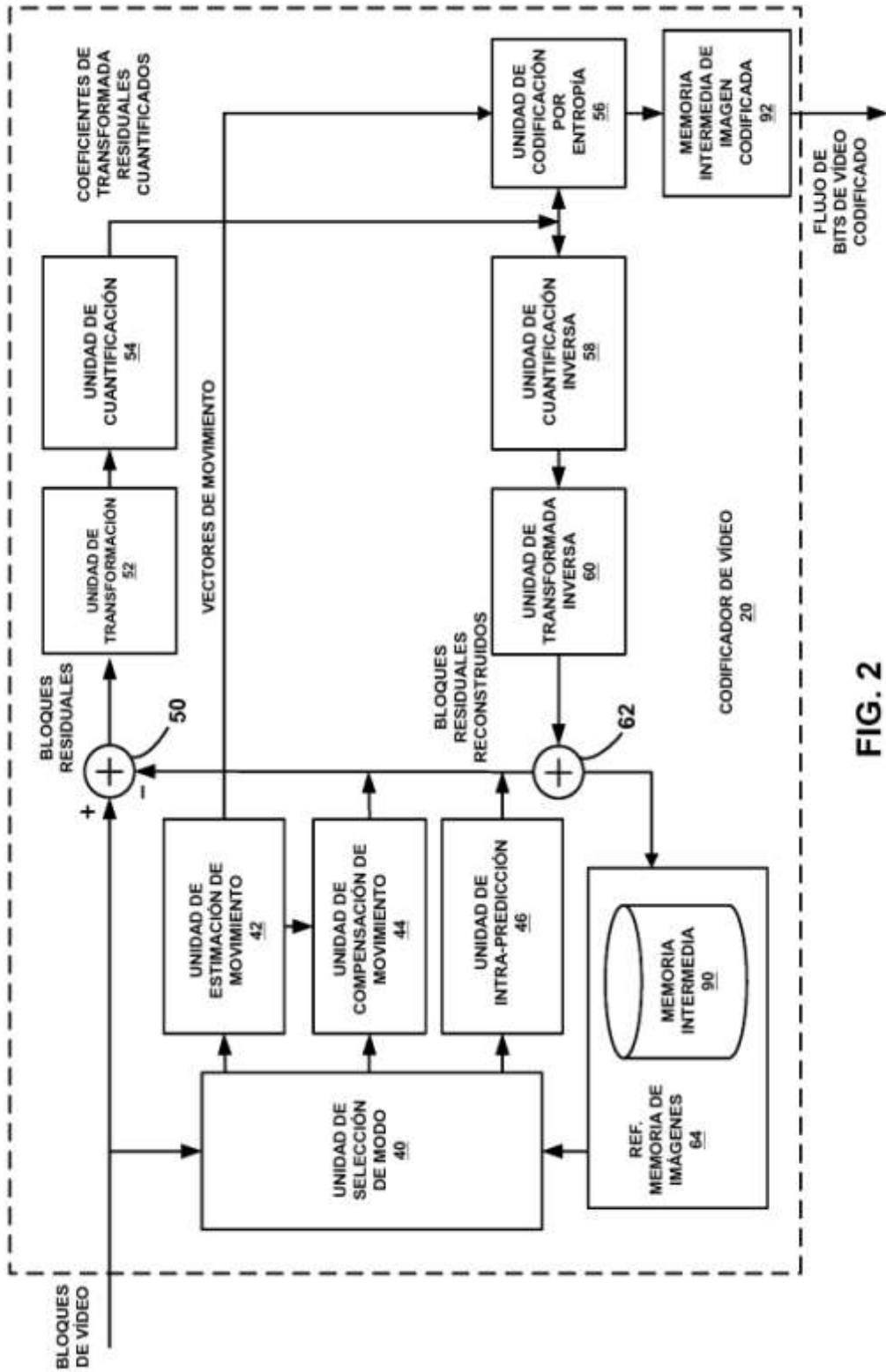


FIG. 2

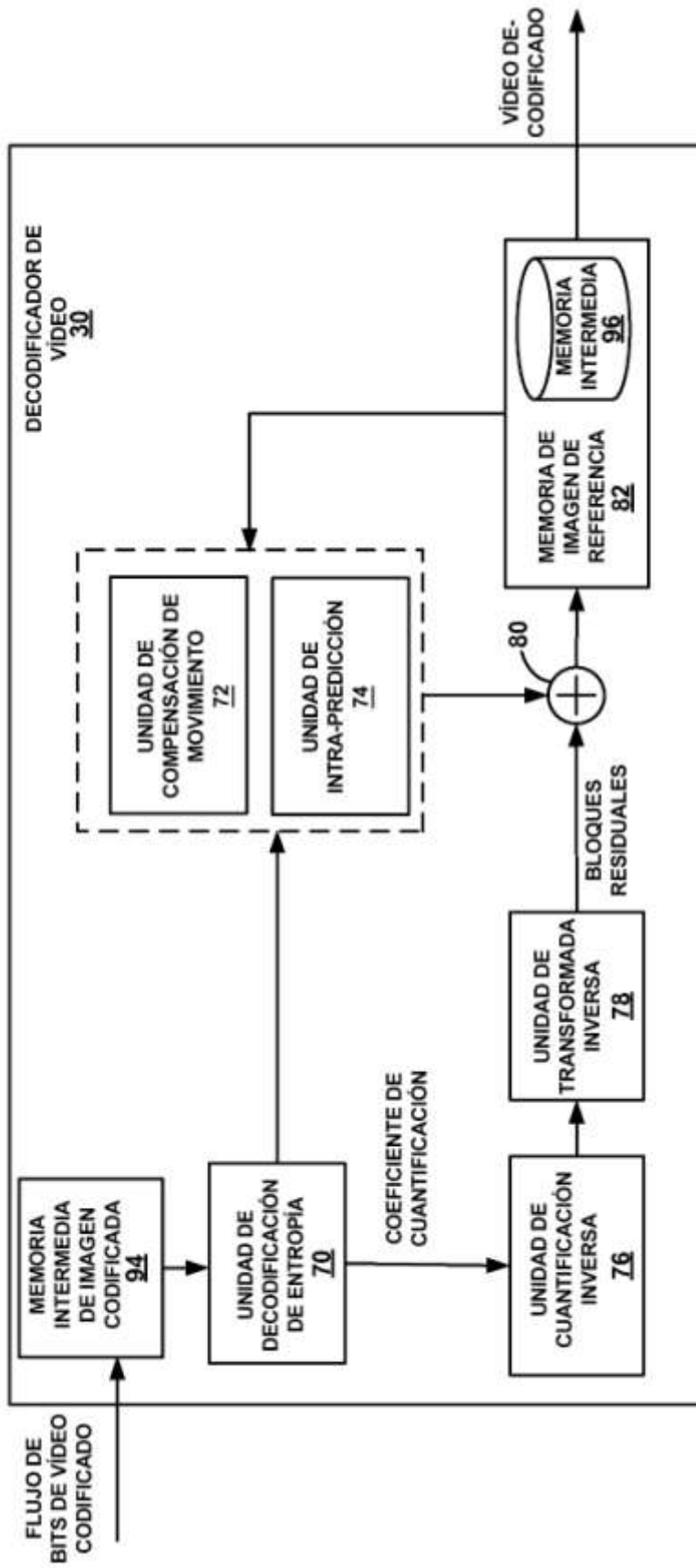


FIG. 3

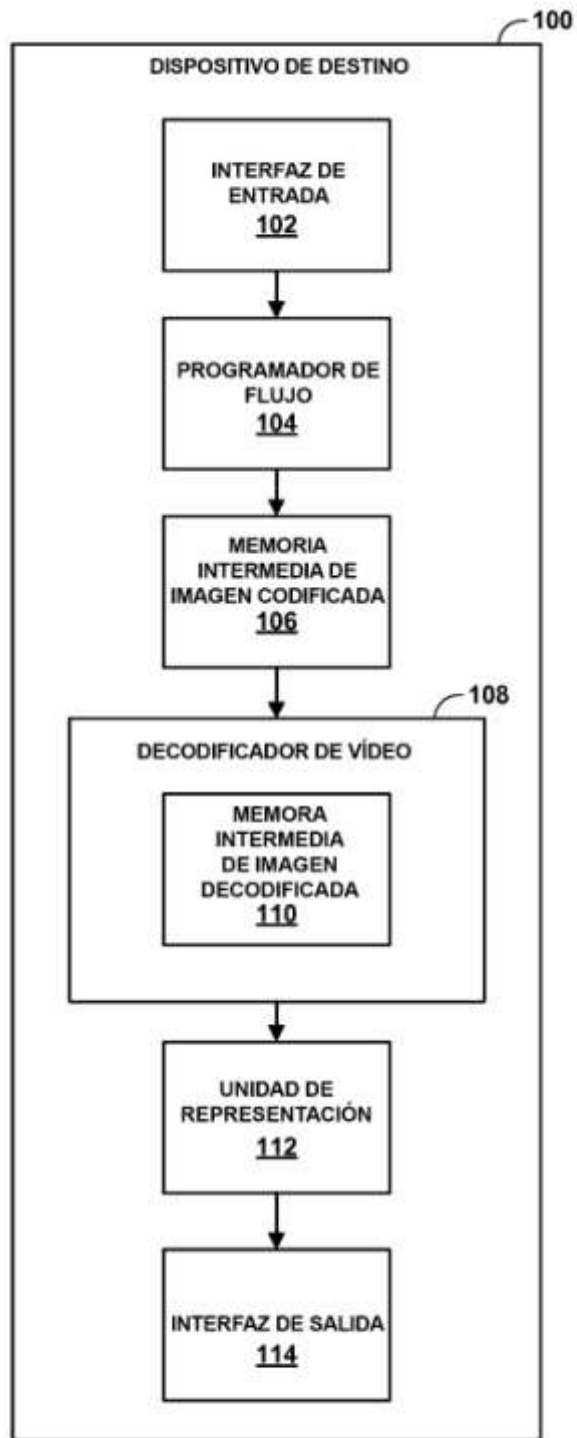


FIG. 4

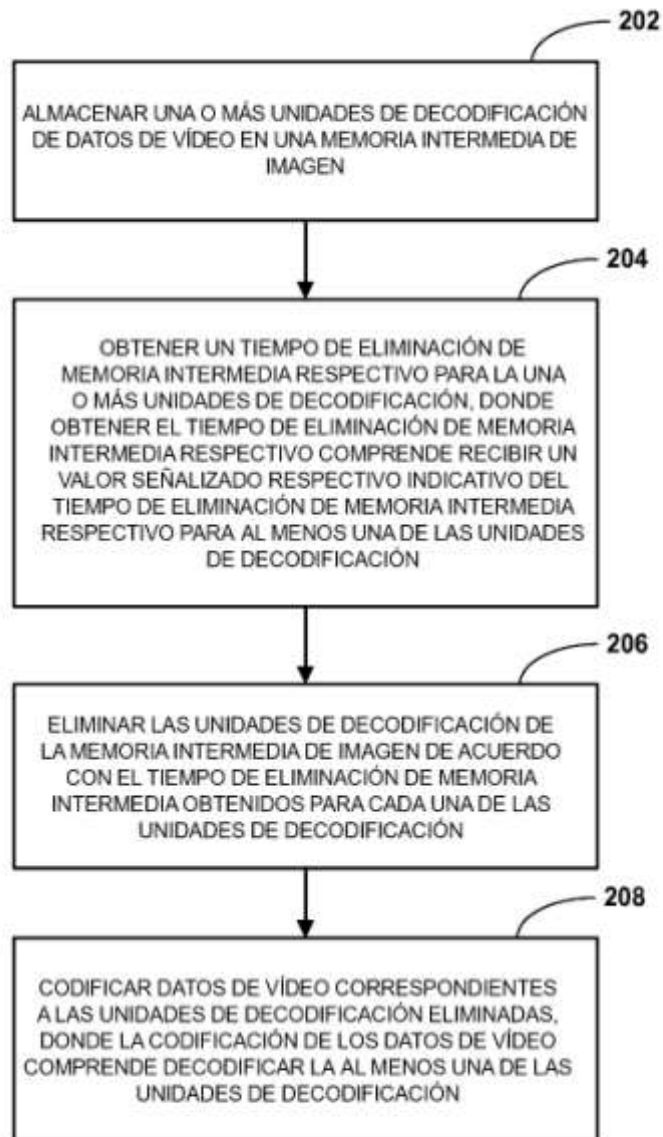


FIG. 5

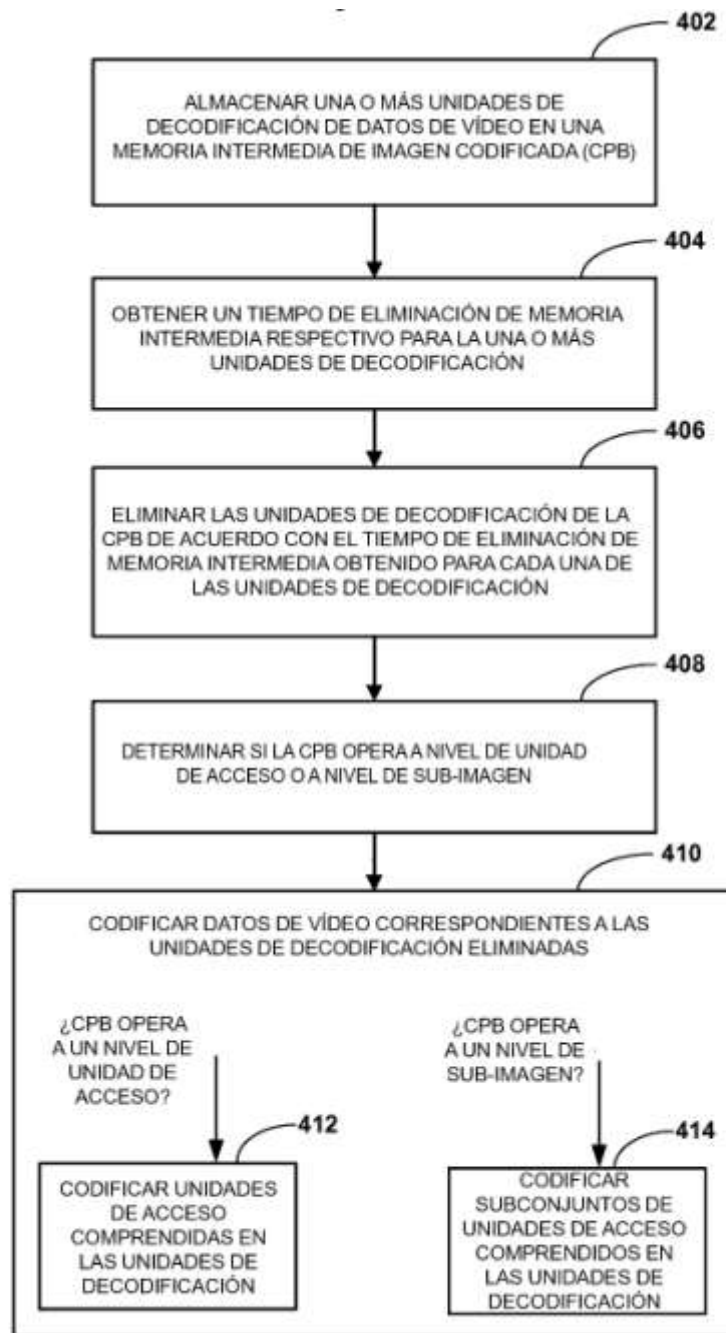


FIG. 6

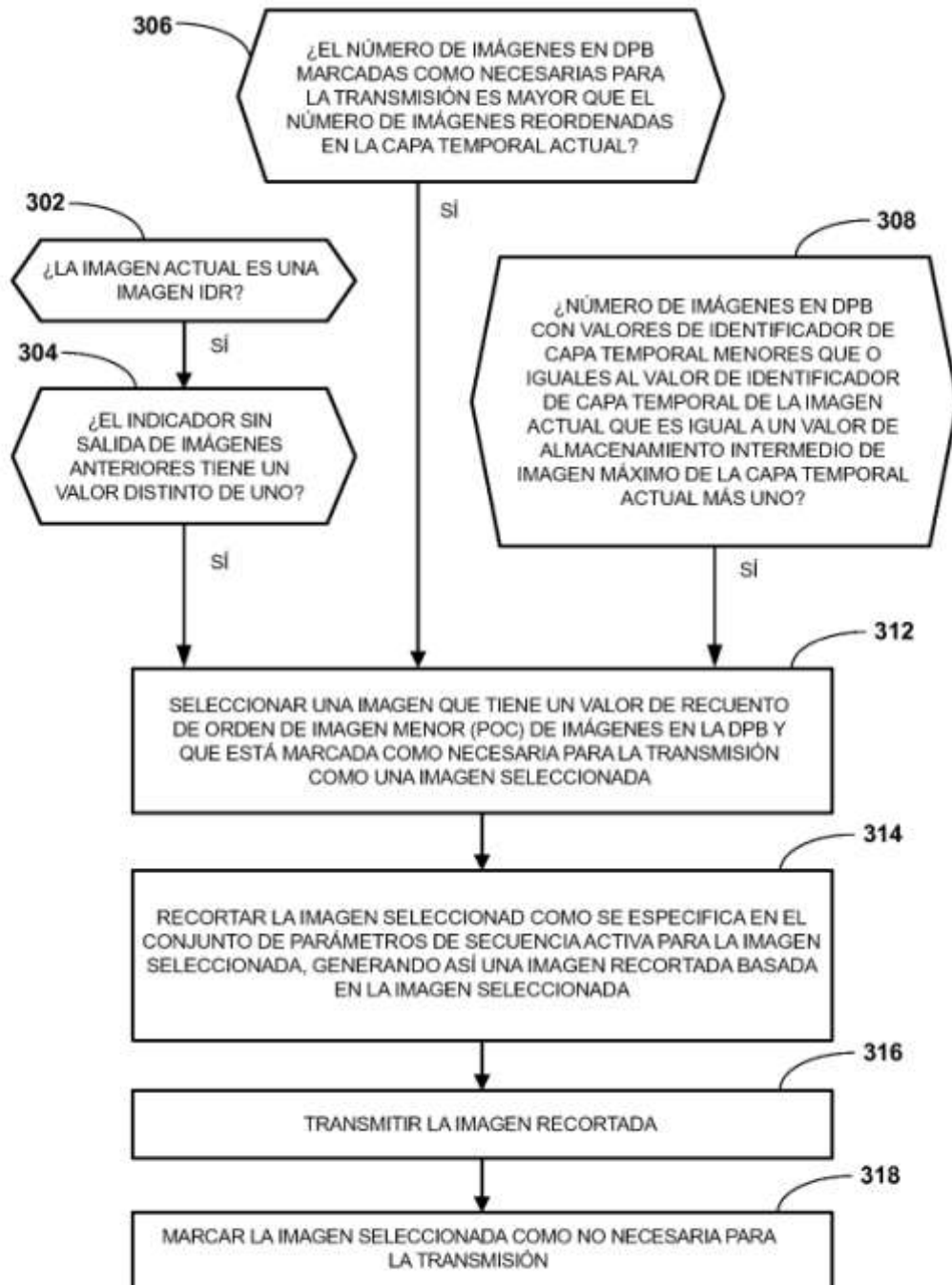


FIG. 7