

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **3 014 003**

51 Int. Cl.:

H04N 19/105 (2014.01)

H04N 19/58 (2014.01)

H04N 19/70 (2014.01)

H04N 19/174 (2014.01)

H04N 19/176 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **16.08.2019** **PCT/US2019/046929**

87 Fecha y número de publicación internacional: **20.02.2020** **WO20037274**

96 Fecha de presentación y número de la solicitud europea: **16.08.2019** **E 19850149 (6)**

97 Fecha y número de publicación de la concesión europea: **27.11.2024** **EP 3831055**

54 Título: **Gestión de imágenes de referencia en codificación de vídeo**

30 Prioridad:

17.08.2018 US 201862719360 P

45 Fecha de publicación y mención en BOPI de la
traducción de la patente:
16.04.2025

73 Titular/es:

HUAWEI TECHNOLOGIES CO., LTD. (100.00%)
Huawei Administration Building Bantian
Longgang District Shenzhen Guangdong 518129,
CN

72 Inventor/es:

WANG, YE-KUI y
HENDRY, FNU

74 Agente/Representante:

PONS ARIÑO, Ángel

ES 3 014 003 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

Gestión de imágenes de referencia en codificación de vídeo

5 Campo técnico

La presente divulgación está relacionada en general con técnicas para la gestión de imágenes de referencia en la codificación de vídeo. Más específicamente, esta divulgación describe técnicas para la construcción de listas de imágenes de referencia y el marcado de imágenes de referencia.

10

Antecedentes

La cantidad de datos de vídeo necesarios para representar incluso un vídeo relativamente corto puede ser considerable, lo que puede dar lugar a dificultades cuando los datos han de transmitirse en continuo o comunicarse de otro modo a través de una red de comunicaciones con capacidad de ancho de banda limitada. Por lo tanto, los datos de vídeo generalmente se comprimen antes de comunicarse a través de las redes de telecomunicaciones modernas. El tamaño de un vídeo también podría ser un problema cuando el vídeo se almacena en un dispositivo de almacenamiento, porque los recursos de memoria son limitados. Los dispositivos de compresión de vídeo a menudo usan *software y/o hardware* en la fuente para codificar los datos de vídeo antes de la transmisión o el almacenamiento, lo que reduce la cantidad de datos necesarios para representar imágenes de vídeo digital. Luego, los datos comprimidos son recibidos en el destino por un dispositivo de descompresión de vídeo que decodifica los datos de vídeo. Con recursos de la red limitados y demandas cada vez mayores de mayor calidad de vídeo, son deseables técnicas mejoradas de compresión y descompresión que mejoren la relación de compresión con poco o ningún sacrificio en la calidad de imagen.

25

HANNUKSELA (NOKIA) M M ET AL, "AHG21: On reference picture list construction and reference picture marking", no. JCTVC-G643, (20111124), 7. JCT-VC MEETING; 20111121 - 20111130; GENEVA; (JOINT COLLABORATIVE TEAM ON VIDEO CODING OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16), URL: http://phenix.int-evry.fr/jct/doc_end_user/documents/7_Geneva/wg11/JCTVC-G643-v3.zip JCTVC-G643r2.doc, (20111124), XP030230939, divulga un enfoque para utilizar directamente tres listas de imágenes de referencia, lista 0 de imágenes de referencia, lista 1 de imágenes de referencia y una lista de imágenes de referencia libre.

30

HANNUKSELA (NOKIA) M M ET AL, "AHG21: On reference picture list construction and reference picture marking", no. JCTVC-G643, (20111124), 7. JCT-VC MEETING; 20111121 - 20111130; GENEVA; (JOINT COLLABORATIVE TEAM ON VIDEO CODING OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16), URL: http://phenix.int-evry.fr/jct/doc_end_user/documents/7_Geneva/wg11/JCTVC-G643-v3.zip JCTVC-F803_d5_RPS_r2_with _JCTVC-G641r2_reduced.doc, (20111124), XP030230935, divulga un enfoque para utilizar directamente tres listas de imágenes de referencia, lista 0 de imágenes de referencia, lista 1 de imágenes de referencia y una lista de imágenes de referencia libre.

35

El documento US 2013/114742 A1 divulga un proceso de marcado de imágenes de referencia y un proceso de gestión de listas de imágenes de referencia.

40 Compendio

45

El objetivo de la presente invención es que un primer aspecto está relacionado con un método para decodificar un flujo de bits de vídeo codificado. El método comprende analizar un encabezado de porción de una porción (*slice*) actual representada en el flujo de bits de vídeo codificado, en donde el encabezado de porción comprende una estructura de lista de imágenes de referencia; obtener, sobre la base de la estructura de lista de imágenes de referencia, una lista de imágenes de referencia de la porción actual; y obtener, sobre la base de la lista de imágenes de referencia, al menos un bloque reconstruido de la porción actual. Este objetivo se resuelve mediante las reivindicaciones independientes adjuntas, y en las reivindicaciones dependientes adjuntas se enumeran realizaciones y mejoras adicionales de la invención. En adelante, hasta la "breve descripción de los dibujos", las expresiones como "... aspecto según la invención", "según la invención" o "la presente invención" se refieren a la enseñanza técnica de la realización más amplia como se reivindica con las reivindicaciones independientes. Los términos como "implementación", "diseño", "opcionalmente", "preferiblemente", "escenario", "aspecto" o similares se refieren a realizaciones adicionales como se reivindica, y los términos y expresiones como "ejemplo", "... aspecto según un ejemplo", "la divulgación describe" o "la divulgación" describen la enseñanza técnica que se refiere a la comprensión de la invención o sus realizaciones, que, sin embargo, no se reivindica como tal. Para lograr el objetivo anterior, se adoptan las soluciones técnicas definidas en las reivindicaciones adjuntas. La invención se define en las reivindicaciones independientes. En las reivindicaciones dependientes se dan a conocer características adicionales de la invención. En lo que sigue, las partes de la descripción y los dibujos que se refieren a realizaciones que no están abarcadas por las reivindicaciones no se presentan como realizaciones de la invención, sino como ejemplos útiles para comprender la invención.

50

El método proporciona técnicas que simplifican y hacen más eficaz la señalización de las listas de imágenes de referencia. Por lo tanto, se mejora el proceso de codificación general.

Breve descripción de los dibujos

5 Para una comprensión más completa de esta divulgación, se hace referencia ahora a la breve descripción siguiente, considerada en conexión con los dibujos adjuntos y la descripción detallada, en donde los números de referencia iguales representan partes iguales.

10 La Figura 1 es un diagrama de bloques que ilustra un sistema de codificación ejemplar que utiliza técnicas de predicción bilateral.

La Figura 2 es un diagrama de bloques que ilustra un codificador de vídeo ejemplar que implementa técnicas de predicción bilateral.

15 La Figura 3 es un diagrama de bloques que ilustra un ejemplo de un decodificador de vídeo que implementa técnicas de predicción bilateral.

La Figura 4 es un diagrama esquemático que ilustra un conjunto de imágenes de referencia (RPS, por sus siglas en inglés) que tiene una imagen con entradas en todos los subconjuntos del RPS.

20 La Figura 5 es una realización de un método de decodificación de un flujo de bits de vídeo codificado.

La Figura 6 es un diagrama esquemático de un dispositivo de codificación de vídeo.

25 La Figura 7 es un diagrama esquemático de una realización de un medio para la codificación.

En la siguiente descripción, las características que en el compendio anterior de la invención se han marcado como "no reivindicadas" también deben entenderse en adelante, cuando se describen y se explican con referencia a los dibujos, como "no reivindicadas" o "que no forman parte de la invención".

30 Descripción detallada

La Figura 1 es un diagrama de bloques que ilustra un sistema 10 de codificación ejemplar que utiliza técnicas de codificación de vídeo como se describe en la presente memoria. Como se muestra en la Figura 1, el sistema 10 de codificación incluye un dispositivo 12 de origen que proporciona datos de vídeo codificados que han de ser decodificados en un momento posterior por un dispositivo 14 de destino. En particular, el dispositivo 12 de origen proporciona los datos de vídeo al dispositivo 14 de destino a través de un medio 16 legible por ordenador. El dispositivo 12 de origen y el dispositivo 14 de destino comprenden cualquiera de una amplia gama de dispositivos, que incluyen ordenadores de escritorio, ordenadores tipo *notebook* (por ejemplo, ordenador portátil), ordenadores tipo tableta, módulos de conexión, aparatos telefónicos tales como los, así llamados, teléfonos "inteligentes", las, así llamadas, "pizarras electrónicas", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivos de transmisión en continuo de vídeo o similares. En algunos casos, el dispositivo 12 de origen y el dispositivo 14 de destino están equipados para la comunicación inalámbrica.

45 El dispositivo 14 de destino recibe los datos de vídeo codificados que se han de decodificar a través del medio 16 legible por ordenador. El medio 16 legible por ordenador comprende cualquier tipo de medio o dispositivo capaz de mover los datos de vídeo codificados del dispositivo 12 de origen al dispositivo 14 de destino. En un ejemplo, el medio 16 legible por ordenador comprende un medio de comunicación para posibilitar que el dispositivo 12 de origen transmita datos de vídeo codificados directamente al dispositivo 14 de destino en tiempo real. Los datos de vídeo codificados se modulan según un estándar de comunicación, tal como un protocolo de comunicación inalámbrica, y se transmiten al dispositivo 14 de destino. El medio de comunicación comprende cualquier medio de comunicación inalámbrica o por cable, tal como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión físicas. El medio de comunicación forma parte de una red basada en paquetes, tal como una red de área local, una red de área amplia o una red global tal como Internet. El medio de comunicación incluye enrutadores, conmutadores, estaciones base o cualquier otro equipo que sea útil para facilitar la comunicación del dispositivo 12 de origen al dispositivo 14 de destino.

En algunos ejemplos, los datos codificados se envían desde una interfaz 22 de salida a un dispositivo de almacenamiento. De manera similar, se accede a los datos codificados desde el dispositivo de almacenamiento mediante una interfaz de entrada. El dispositivo de almacenamiento incluye cualquiera de diversos medios de almacenamiento de datos distribuidos o de acceso local, tales como un disco duro, discos Blu-ray, discos de vídeo digital (DVD), Memorias de Sólo Lectura de Disco Compacto (CD-ROM), memoria *flash*, memoria volátil o no volátil, o cualquier otro medio de almacenamiento digital adecuado para almacenar datos de vídeo codificados. En otro ejemplo, el dispositivo de almacenamiento corresponde a un servidor de archivos u otro dispositivo de almacenamiento intermedio que almacene el vídeo codificado generado por el dispositivo 12 de origen. El dispositivo 14 de destino accede a datos de vídeo almacenados desde el dispositivo de almacenamiento a través de transmisión en continuo o descarga. El servidor de archivos es cualquier tipo de servidor capaz de almacenar datos de vídeo codificados y

transmitir esos datos de vídeo codificados al dispositivo 14 de destino. Los servidores de archivos ejemplares incluyen un servidor web (por ejemplo, para un sitio web), un servidor de protocolo de transferencia de archivos (FTP, por sus siglas en inglés), dispositivos de almacenamiento conectados a la red (NAS, por sus siglas en inglés) o una unidad de disco local. El dispositivo 14 de destino accede a los datos de vídeo codificados a través de cualquier conexión de datos estándar, incluida una conexión a Internet. Esto incluye un canal inalámbrico (por ejemplo, una conexión Wi-Fi), una conexión por cable (por ejemplo, línea de abonado digital (DSL, por sus siglas en inglés), módem por cable, etc.) o una combinación de ambos que sea adecuada para acceder a datos de vídeo codificados almacenados en un servidor de archivos. La transmisión de datos de vídeo codificados desde el dispositivo de almacenamiento es una transmisión en continuo, una transmisión de descarga o una combinación de las mismas.

Las técnicas de esta divulgación no se limitan necesariamente a aplicaciones o configuraciones inalámbricas. Las técnicas se aplican a la codificación de vídeo en apoyo de cualquiera de diversas aplicaciones multimedia, tales como radiodifusiones de televisión por aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de vídeo en continuo por Internet, tales como transmisión dinámica adaptativa en continuo sobre HTTP (DASH, por sus siglas en inglés), vídeo digital codificado en un medio de almacenamiento de datos, decodificación de vídeo digital almacenado en un medio de almacenamiento de datos u otras aplicaciones. En algunos ejemplos, el sistema 10 de codificación está configurado para admitir transmisión de vídeo unidireccional o bidireccional para admitir aplicaciones tales como transmisión en continuo de vídeo, reproducción de vídeo, radiodifusión de vídeo y/o videotelefonía.

En el ejemplo de la Figura 1, el dispositivo 12 de origen incluye una fuente 18 de vídeo, un codificador 20 de vídeo y la interfaz 22 de salida. El dispositivo 14 de destino incluye una interfaz 28 de entrada, un decodificador 30 de vídeo y un dispositivo 32 de visualización. Según esta divulgación, el codificador 20 de vídeo del dispositivo 12 de origen y/o el decodificador 30 de vídeo del dispositivo 14 de destino están configurados para aplicar las técnicas para la codificación de vídeo. En otros ejemplos, un dispositivo de origen y un dispositivo de destino incluyen otros componentes o disposiciones. El dispositivo 12 de origen recibe datos de vídeo desde una fuente de vídeo externa, tal como una cámara externa. Asimismo, el dispositivo 14 de destino interactúa con un dispositivo de visualización externo, en lugar de incluir un dispositivo de visualización integrado.

El sistema 10 de codificación ilustrado de la Figura 1 es simplemente un ejemplo. Las técnicas para la codificación de vídeo se realizan mediante cualquier dispositivo de codificación y/o decodificación de vídeo digital. Aunque las técnicas de esta divulgación generalmente se realizan mediante un dispositivo de codificación de vídeo, las técnicas también se realizan mediante un codificador/decodificador de vídeo, normalmente denominado "CODEC". Además, las técnicas de esta divulgación también se realizan mediante un preprocesador de vídeo. El codificador y/o el decodificador de vídeo son una unidad de procesamiento de gráficos (GPU, por sus siglas en inglés) o un dispositivo similar.

El dispositivo 12 de origen y el dispositivo 14 de destino son simplemente ejemplos de tales dispositivos de codificación en los que el dispositivo 12 de origen genera datos de vídeo codificados para transmitirlos al dispositivo 14 de destino. En algunos ejemplos, el dispositivo 12 de origen y el dispositivo 14 de destino funcionan de manera sustancialmente simétrica, de modo que los dispositivos 12, 14 de origen y de destino incluyen cada uno componentes de codificación y decodificación de vídeo. Por lo tanto, el sistema 10 de codificación admite transmisión de vídeo unidireccional o bidireccional entre dispositivos 12, 14 de vídeo, por ejemplo, para transmisión en continuo de vídeo, reproducción de vídeo, radiodifusión de vídeo o videotelefonía.

La fuente 18 de vídeo del dispositivo 12 de origen incluye un dispositivo de captura de vídeo, tal como una cámara de vídeo, un archivo de vídeo que contenga vídeo capturado previamente y/o una interfaz de alimentación de vídeo para recibir vídeo de un proveedor de contenido de vídeo. Como alternativa adicional, la fuente 18 de vídeo genera datos basados en gráficos de ordenador como la fuente de vídeo, o una combinación de vídeo en vivo, vídeo archivado y vídeo generado por ordenador.

En algunos casos, cuando la fuente 18 de vídeo es una cámara de vídeo, el dispositivo 12 de origen y el dispositivo 14 de destino forman unos, así llamados, teléfonos con cámara o videoteléfonos. Sin embargo, como se mencionó anteriormente, las técnicas descritas en esta divulgación son aplicables a la codificación de vídeo en general y se aplican a aplicaciones inalámbricas y/o por cable. En cada caso, el vídeo capturado, precapturado o generado por ordenador se codifica mediante el codificador 20 de vídeo. La información de vídeo codificada se envía luego mediante la interfaz 22 de salida a un medio 16 legible por ordenador.

El medio 16 legible por ordenador incluye medios transitorios, tales como una radiodifusión inalámbrica o una transmisión por red por cable, o medios de almacenamiento (es decir, medios de almacenamiento no transitorios), tales como un disco duro, una unidad *flash*, un disco compacto, un disco de vídeo digital, un disco Blu-ray u otro medio legible por ordenador. En algunos ejemplos, un servidor de red (no mostrado) recibe datos de vídeo codificados desde el dispositivo 12 de origen y proporciona los datos de vídeo codificados al dispositivo 14 de destino, por ejemplo, a través de una transmisión por red. De manera similar, un dispositivo informático de una instalación de producción de medios, tal como una instalación de impresión de discos, recibe datos de vídeo codificados desde el dispositivo 12 de origen y produce un disco que contiene los datos de vídeo codificados. Por lo tanto, se entiende que el medio 16

legible por ordenador incluye uno o más medios legibles por ordenador de diversas formas, en diversos ejemplos.

La interfaz 28 de entrada del dispositivo 14 de destino recibe información del medio 16 legible por ordenador. La información del medio 16 legible por ordenador incluye información de sintaxis definida por el codificador 20 de vídeo, que también es usada por el decodificador 30 de vídeo, que incluye elementos de sintaxis que describen características y/o procesamiento de bloques y otras unidades codificadas, por ejemplo, grupo de imágenes (GOP, por sus siglas en inglés). El dispositivo 32 de visualización muestra los datos de vídeo decodificados a un usuario y comprende cualquiera de diversos dispositivos de visualización tales como un tubo de rayos catódicos (CRT, por sus siglas en inglés), una pantalla de cristal líquido (LCD, por sus siglas en inglés), una pantalla de plasma, una pantalla de diodos orgánicos emisores de luz (OLED, por sus siglas en inglés) u otro tipo de dispositivo de visualización.

El codificador 20 de vídeo y el decodificador 30 de vídeo funcionan según el estándar de Codificación de Vídeo de Alta Eficiencia (HEVC, por sus siglas en inglés) actualmente en desarrollo y se ajustan al Modelo de Prueba HEVC (HM, por sus siglas en inglés). Otros estándares patentados o de la industria incluyen el estándar H.264 del Sector de Normalización de las Telecomunicaciones de la Unión Internacional de Telecomunicaciones (ITU-T, por sus siglas en inglés), también denominado Grupo de Expertos en Imágenes en Movimiento (MPEG, por sus siglas en inglés)-4, Parte 10, Codificación de Vídeo Avanzada (AVC, por sus siglas en inglés), H.265/HEVC o extensiones de tales estándares. Otros ejemplos de estándares de codificación de vídeo incluyen MPEG-2 y H.263 de ITU-T. Aunque no se muestra en la Figura 1, en algunos aspectos, el codificador 20 de vídeo y el decodificador 30 de vídeo están integrados cada uno con un codificador y decodificador de audio e incluyen unidades de multiplexor-demultiplexor (MUX-DEMUX) apropiadas, u otros *hardware* y *software*, para manejar la codificación tanto de audio como de vídeo en un flujo de datos común o en flujos de datos separados. Si corresponde, las unidades MUX-DEMUX se ajustan al protocolo multiplexor H.223 de ITU u otros protocolos, tales como el protocolo de datagramas de usuario (UDP, por sus siglas en inglés).

El codificador 20 de vídeo y el decodificador 30 de vídeo se implementan cada uno como cualquiera de diversos circuitos codificadores adecuados, tales como uno o más microprocesadores, procesadores de señales digitales (DSP, por sus siglas en inglés), circuitos integrados de aplicación específica (ASIC, por sus siglas en inglés), agrupaciones de puertas programables de campo (FPGA, por sus siglas en inglés), lógica discreta, *software*, *hardware*, *firmware* o cualesquiera combinaciones de los mismos. Cuando las técnicas se implementan parcialmente en *software*, un dispositivo almacena instrucciones para el *software* en un medio adecuado legible por ordenador no transitorio y ejecuta las instrucciones en *hardware* usando uno o más procesadores para realizar las técnicas de esta divulgación. El codificador 20 de vídeo y el decodificador 30 de vídeo se incluyen cada uno en uno o más codificadores o decodificadores, cualquiera de los cuales está integrado como parte de un codificador/decodificador (CODEC) combinado en un dispositivo respectivo. Un dispositivo que incluye un codificador 20 de vídeo y/o un decodificador 30 de vídeo comprende un circuito integrado, un microprocesador y/o un dispositivo de comunicación inalámbrica, tal como un teléfono celular.

La Figura 2 es un diagrama de bloques que ilustra un ejemplo de codificador 20 de vídeo que implementa técnicas de codificación de vídeo. El codificador 20 de vídeo realiza intracodificación e intercodificación de bloques de vídeo dentro de porciones de vídeo. La intracodificación se basa en la predicción espacial para reducir o eliminar la redundancia espacial en el vídeo dentro de un fotograma o imagen de vídeo dado. La intercodificación se basa en la predicción temporal para reducir o eliminar la redundancia temporal en el vídeo dentro de fotogramas o imágenes adyacentes de una secuencia de vídeo. El intramodo (modo I) se refiere a cualquiera de varios modos de codificación basados en el espacio. Los intermodos, tales como la predicción unidireccional (también conocida como unipredicción) (modo P) o la predicción bidireccional (también conocida como bipredicción) (modo B), se refieren a cualquiera de varios modos de codificación basados en el tiempo.

Como se muestra en la Figura 2, el codificador 20 de vídeo recibe un bloque de vídeo actual dentro de un fotograma de vídeo que ha de ser codificado. En el ejemplo de la Figura 2, el codificador 20 de vídeo incluye la unidad 40 de selección de modo, la memoria 64 de fotogramas de referencia, el sumador 50, la unidad 52 de procesamiento de transformada, la unidad 54 de cuantificación y la unidad 56 de codificación de entropía. La unidad 40 de selección de modo, a su vez, incluye la unidad 44 de compensación de movimiento, la unidad 42 de estimación de movimiento, la unidad 46 de intrapredicción (también conocida como intrapredicción) y la unidad 48 de partición. Para la reconstrucción de bloques de vídeo, el codificador 20 de vídeo también incluye la unidad 58 de cuantificación inversa, la unidad 60 de transformada inversa y el sumador 62. También se incluye un filtro de desbloqueo (no mostrado en la Figura 2) para filtrar los límites del bloque para eliminar los artefactos de bloqueo del vídeo reconstruido. Si se desea, el filtro de desbloqueo normalmente filtraría la salida del sumador 62. También se usan filtros adicionales (en bucle o postbucle) además del filtro de desbloqueo. Tales filtros no se muestran por brevedad, pero, si se desean, filtran la salida del sumador 50 (como un filtro en bucle).

Durante el proceso de codificación, el codificador 20 de vídeo recibe un fotograma o porción de vídeo que se ha de codificar. El fotograma o porción se divide en múltiples bloques de vídeo. La unidad 42 de estimación de movimiento y la unidad 44 de compensación de movimiento realizan una codificación interpredictiva del bloque de vídeo recibido en relación con uno o más bloques en uno o más fotogramas de referencia para proporcionar una predicción temporal. La unidad 46 de intrapredicción realiza alternativamente una codificación intrapredictiva del bloque de vídeo recibido

en relación con uno o más bloques vecinos en el mismo fotograma o porción que el bloque que se ha de codificar para proporcionar predicción espacial. El codificador 20 de vídeo realiza múltiples pasadas de codificación, por ejemplo, para seleccionar un modo de codificación apropiado para cada bloque de datos de vídeo.

- 5 Además, la unidad 48 de partición divide bloques de datos de vídeo en subbloques, en función de la evaluación de esquemas de partición anteriores en pasadas de codificación anteriores. La unidad 48 de partición divide inicialmente un fotograma o porción en unidades de codificación más grandes (LCU, por sus siglas en inglés) y divide cada una de las LCU en unidades de subcodificación (sub-CU, por sus siglas en inglés) en función del análisis de distorsión de tasa (por ejemplo, optimización de distorsión de tasa). La unidad 40 de selección de modo produce además una estructura de datos de árbol cuádruple indicativa de la partición de una LCU en sub-CU. Las CU de nodo de hoja del árbol cuádruple incluyen una o más unidades de predicción (PU, por sus siglas en inglés) y una o más unidades de transformada (TU, por sus siglas en inglés).

- 15 La presente divulgación usa el término "bloque" para referirse a cualquiera de una CU, PU o TU, en el contexto de HEVC, o estructuras de datos similares en el contexto de otros estándares (por ejemplo, macrobloques y subbloques de los mismos en H.264/AVC). Una CU incluye un nodo de codificación, PU y TU asociadas con el nodo de codificación. Un tamaño de la CU corresponde a un tamaño del nodo de codificación y tiene forma cuadrada. El tamaño de la CU varía desde 8×8 píxeles hasta el tamaño del bloque de árbol con un máximo de 64×64 píxeles o más. Cada CU contiene una o más PU y una o más TU. Los datos de sintaxis asociados con una CU describen, por ejemplo, la división de la CU en una o más PU. Los modos de división difieren entre si la CU está codificada en modo directo o saltado, codificada en modo intrapredicción o codificada en modo interpredicción. Las PU se dividen para que no tengan forma cuadrada. Los datos de sintaxis asociados con una CU también describen la división de la CU en una o más TU según un árbol cuádruple. Una TU puede tener forma cuadrada o no cuadrada (por ejemplo, rectangular).

- 25 La unidad 40 de selección de modo selecciona uno de los modos de codificación, intra- o inter-, por ejemplo, en función de los resultados de error, y proporciona el bloque intracodificado o intercodificado resultante al sumador 50 para generar datos de bloque residual y al sumador 62 para reconstruir el bloque codificado para su uso como fotograma de referencia. La unidad 40 de selección de modo también proporciona elementos de sintaxis, tales como vectores de movimiento, indicadores intramodo, información de partición y otra información de sintaxis similar, a la unidad 56 de codificación de entropía.

- 35 La unidad 42 de estimación de movimiento y la unidad 44 de compensación de movimiento están altamente integradas, pero se ilustran por separado con fines conceptuales. La estimación de movimiento, realizada por la unidad 42 de estimación de movimiento, es el proceso de generar vectores de movimiento, que estiman el movimiento para bloques de vídeo. Un vector de movimiento indica el desplazamiento de una PU de un bloque de vídeo dentro de un fotograma o imagen de vídeo actual en relación con un bloque predictivo dentro de un fotograma de referencia (u otra unidad codificada) en relación con el bloque actual que se está codificando dentro del fotograma actual (u otra unidad codificada). Un bloque predictivo es un bloque del que se comprueba que coincide estrechamente con el bloque que se ha de codificar, en términos de diferencia de píxeles, que se determina mediante la suma de la diferencia absoluta (SAD, por sus siglas en inglés), la suma de la diferencia cuadrática (SSD, por sus siglas en inglés) u otras métricas de diferencia. En algunos ejemplos, el codificador 20 de vídeo calcula valores para posiciones de píxeles de subenteros de imágenes de referencia almacenadas en la memoria 64 de fotogramas de referencia. El codificador 20 de vídeo interpola valores de posiciones de un cuarto de píxel, posiciones de un octavo de píxel u otras posiciones fraccionarias de píxel de la imagen de referencia. Por lo tanto, la unidad 42 de estimación de movimiento realiza una búsqueda de movimiento en relación con las posiciones de píxeles completos y posiciones de píxeles fraccionarios y emite un vector de movimiento con precisión de píxeles fraccionarios.

- 45 La unidad 42 de estimación de movimiento calcula un vector de movimiento para una PU de un bloque de vídeo en una porción intercodificada comparando la posición de la PU con la posición de un bloque predictivo de una imagen de referencia. La imagen de referencia se selecciona de una primera lista de imágenes de referencia (Lista 0) o una segunda lista de imágenes de referencia (Lista 1), cada una de las cuales identifica una o más imágenes de referencia almacenadas en la memoria 64 de fotogramas de referencia. La unidad 42 de estimación de movimiento envía el vector de movimiento calculado a la unidad 56 de codificación de entropía y a la unidad 44 de compensación de movimiento.

- 55 La compensación de movimiento, realizada por la unidad 44 de compensación de movimiento, implica buscar o generar el bloque predictivo en función del vector de movimiento determinado por la unidad 42 de estimación de movimiento. Nuevamente, la unidad 42 de estimación de movimiento y la unidad 44 de compensación de movimiento están integradas funcionalmente, en algunos ejemplos. Al recibir el vector de movimiento para la PU del bloque de vídeo actual, la unidad 44 de compensación de movimiento localiza el bloque predictivo al que apunta el vector de movimiento en una de las listas de imágenes de referencia. El sumador 50 forma un bloque de vídeo residual restando los valores de píxel del bloque predictivo de los valores de píxel del bloque de vídeo actual que se está codificando, formando valores de diferencia de píxel, como se explica posteriormente. En general, la unidad 42 de estimación de movimiento realiza la estimación de movimiento en relación con los componentes de luma, y la unidad 44 de compensación de movimiento usa vectores de movimiento calculados en función de los componentes de luma tanto para los componentes de croma como para los componentes de luma. La unidad 40 de selección de modo también

genera elementos de sintaxis asociados con los bloques de vídeo y la porción de vídeo para su uso por el decodificador 30 de vídeo al decodificar los bloques de vídeo de la porción de vídeo.

La unidad 46 de intrapredicción intrapredice un bloque actual, como una alternativa a la interpretación realizada por la unidad 42 de estimación de movimiento y la unidad 44 de compensación de movimiento, como se describió anteriormente. En particular, la unidad 46 de intrapredicción determina un modo de intrapredicción que se ha de usar para codificar un bloque actual. En algunos ejemplos, la unidad 46 de intrapredicción codifica un bloque actual usando diversos modos de intrapredicción, por ejemplo, durante pasadas de codificación separadas, y la unidad 46 de intrapredicción (o la unidad 40 de selección de modo, en algunos ejemplos) selecciona un modo de intrapredicción apropiado para su uso a partir de los modos probados.

La unidad 46 de intrapredicción calcula valores de distorsión de tasa usando un análisis de distorsión de tasa para los diversos modos de intrapredicción probados y selecciona el modo de intrapredicción que tenga las mejores características de distorsión de tasa entre los modos probados. El análisis de distorsión de tasa generalmente determina una cantidad de distorsión (o error) entre un bloque codificado y un bloque original sin codificar que fue codificado para producir el bloque codificado, así como una tasa de bits (es decir, una cantidad de bits) utilizada para producir el bloque codificado. La unidad 46 de intrapredicción calcula relaciones a partir de las distorsiones y tasas para los diversos bloques codificados para determinar qué modo de intrapredicción presenta el mejor valor de distorsión de tasa para el bloque.

Además, la unidad 46 de intrapredicción se configura para codificar bloques de profundidad de un mapa de profundidad usando un modo de modelado de profundidad (DMM, por sus siglas en inglés). La unidad 40 de selección de modo determina si un modo DMM disponible produce mejores resultados de codificación que un modo de intrapredicción y los otros modos DMM, por ejemplo, usando optimización de distorsión de tasa (RDO, por sus siglas en inglés). Los datos para una imagen de textura correspondiente a un mapa de profundidad se almacenan en la memoria 64 de fotogramas de referencia. La unidad 42 de estimación de movimiento y la unidad 44 de compensación de movimiento también se configuran para interpretar bloques de profundidad de un mapa de profundidad.

Después de seleccionar un modo de intrapredicción para un bloque (por ejemplo, un modo de intrapredicción convencional o uno de los modos DMM), la unidad 46 de intrapredicción proporciona información indicativa del modo de intrapredicción seleccionado para el bloque a la unidad 56 de codificación de entropía. La unidad 56 de codificación de entropía codifica la información que indica el modo de intrapredicción seleccionado. El codificador 20 de vídeo incluye en los datos de configuración de flujo de bits transmitidos, que incluyen una pluralidad de tablas de índices de modo de intrapredicción y una pluralidad de tablas de índices de modo de intrapredicción modificadas (también denominadas tablas de mapeo de palabras código), definiciones de contextos de codificación para diversos bloques, e indicaciones de un modo de intrapredicción más probable, una tabla de índices de modo de intrapredicción y una tabla de índices de modo de intrapredicción modificada para su uso en cada uno de los contextos.

El codificador 20 de vídeo forma un bloque de vídeo residual restando los datos de predicción de la unidad 40 de selección de modo del bloque de vídeo original que se está codificando. El sumador 50 representa el componente o componentes que realizan esta operación de resta.

La unidad 52 de procesamiento de transformada aplica una transformada, tal como una transformada de coseno discreta (DCT, por sus siglas en inglés) o una transformada conceptualmente similar, al bloque residual, produciendo un bloque de vídeo que comprende valores de coeficiente de transformada residual. La unidad 52 de procesamiento de transformada realiza otras transformadas que son conceptualmente similares a la DCT. También podrían usarse transformadas de ondícula, transformadas de enteros, transformadas de subbanda u otros tipos de transformadas.

La unidad 52 de procesamiento de transformada aplica la transformada al bloque residual, produciendo un bloque de coeficientes de transformada residual. La transformada convierte la información residual de un dominio de valor de píxel en un dominio de transformada, tal como un dominio de frecuencia. La unidad 52 de procesamiento de transformada envía los coeficientes de transformada resultantes a la unidad 54 de cuantificación. La unidad 54 de cuantificación cuantifica los coeficientes de transformada para reducir aun más la tasa de bits. El proceso de cuantificación reduce la profundidad de bit asociada con algunos o la totalidad de los coeficientes. El grado de cuantificación se modifica ajustando un parámetro de cuantificación. En algunos ejemplos, la unidad 54 de cuantificación realiza entonces un escaneo de la matriz que incluye los coeficientes de transformada cuantificados. Como alternativa, la unidad 56 de codificación de entropía realiza el escaneo.

Después de la cuantificación, la unidad 56 de codificación de entropía codifica los coeficientes de transformada cuantificados. La unidad 56 de codificación de entropía realiza una codificación de longitud variable adaptativa según el contexto (CAVLC, por sus siglas en inglés), codificación aritmética binaria adaptativa según el contexto (CABAC, por sus siglas en inglés), codificación aritmética binaria adaptativa según el contexto basada en sintaxis (SBAC, por sus siglas en inglés), codificación de entropía de partición de intervalo de probabilidad (PIPE, por sus siglas en inglés) u otra técnica de codificación de entropía. En el caso de la codificación de entropía basada en el contexto, el contexto se basa en bloques vecinos. Después de la codificación de entropía por la unidad 56 de codificación de entropía, el flujo de bits codificado se transmite a otro dispositivo (por ejemplo, el decodificador 30 de vídeo) o se archiva para su

posterior transmisión o recuperación.

La unidad 58 de cuantificación inversa y la unidad 60 de transformada inversa aplican cuantificación inversa y transformación inversa, respectivamente, para reconstruir el bloque residual en el dominio de píxeles, por ejemplo, para uso posterior como bloque de referencia. La unidad 44 de compensación de movimiento calcula un bloque de referencia añadiendo el bloque residual a un bloque predictivo de uno de los fotogramas de la memoria 64 de fotogramas de referencia. La unidad 44 de compensación de movimiento también aplica uno o más filtros de interpolación al bloque residual reconstruido para calcular valores de píxeles subenteros para su uso en la estimación de movimiento. El sumador 62 añade el bloque residual reconstruido al bloque de predicción con compensación de movimiento producido por la unidad 44 de compensación de movimiento para producir un bloque de vídeo reconstruido para su almacenamiento en la memoria 64 de fotogramas de referencia. El bloque de vídeo reconstruido lo usan la unidad 42 de estimación de movimiento y la unidad 44 de compensación de movimiento como un bloque de referencia para intercodificar un bloque en un fotograma de vídeo posterior.

La Figura 3 es un diagrama de bloques que ilustra un ejemplo de decodificador 30 de vídeo que implementa técnicas de codificación de vídeo. En el ejemplo de la Figura 3, el decodificador 30 de vídeo incluye una unidad 70 de decodificación de entropía, una unidad 72 de compensación de movimiento, una unidad 74 de intrapredicción, una unidad 76 de cuantificación inversa, una unidad 78 de transformación inversa, una memoria 82 de fotogramas de referencia y un sumador 80. El decodificador 30 de vídeo puede, en algunos ejemplos, realizar una pasada de decodificación generalmente recíproca a la pasada de codificación descrita con respecto al codificador 20 de vídeo (Figura 2). La unidad 72 de compensación de movimiento genera datos de predicción en función de vectores de movimiento recibidos de la unidad 70 de decodificación de entropía, mientras que la unidad 74 de intrapredicción genera datos de predicción en función de indicadores de modo de intrapredicción recibidos de la unidad 70 de decodificación de entropía.

Durante el proceso de decodificación, el decodificador 30 de vídeo recibe un flujo de bits de vídeo codificado que representa bloques de vídeo de una porción de vídeo codificada y elementos de sintaxis asociados del codificador 20 de vídeo. La unidad 70 de decodificación de entropía del decodificador 30 de vídeo decodifica la entropía del flujo de bits para generar coeficientes cuantificados, vectores de movimiento o indicadores de modo de intrapredicción y otros elementos de sintaxis. La unidad 70 de decodificación de entropía envía los vectores de movimiento y otros elementos de sintaxis a la unidad 72 de compensación de movimiento. El decodificador 30 de vídeo recibe los elementos de sintaxis en el nivel de porción de vídeo y/o el nivel de bloque de vídeo.

Cuando la porción de vídeo se codifica como una porción intracodificada (I), la unidad 74 de intrapredicción genera datos de predicción para un bloque de vídeo de la porción de vídeo actual en función de un modo de intrapredicción señalizado y datos de bloques previamente decodificados del fotograma o imagen actual. Cuando el fotograma de vídeo se codifica como una porción intercodificada (por ejemplo, B, P o GPB), la unidad 72 de compensación de movimiento produce bloques predictivos para un bloque de vídeo de la porción de vídeo actual en función de los vectores de movimiento y otros elementos de sintaxis recibidos de la unidad 70 de decodificación de entropía. Los bloques predictivos se producen a partir de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. El decodificador 30 de vídeo construye las listas de fotogramas de referencia, Lista 0 y Lista 1, usando técnicas de construcción por defecto en función de imágenes de referencia almacenadas en la memoria 82 de fotogramas de referencia.

La unidad 72 de compensación de movimiento determina información de predicción para un bloque de vídeo de la porción de vídeo actual analizando los vectores de movimiento y otros elementos de sintaxis y usa la información de predicción para producir los bloques predictivos para el bloque de vídeo actual que se está decodificando. La unidad 72 de compensación de movimiento usa algunos de los elementos de sintaxis recibidos para determinar un modo de predicción (por ejemplo, intrapredicción o interpredicción) usado para codificar los bloques de vídeo de la porción de vídeo, un tipo de porción de interpredicción (por ejemplo, porción B, porción P o porción GPB), información de construcción para una o más de las listas de imágenes de referencia para la porción, vectores de movimiento para cada bloque de vídeo intercodificado de la porción, estado de interpredicción para cada bloque de vídeo intercodificado de la porción y otra información para decodificar los bloques de vídeo en la porción de vídeo actual.

La unidad 72 de compensación de movimiento también realiza una interpolación basada en filtros de interpolación. La unidad 72 de compensación de movimiento usa filtros de interpolación como los que usa el codificador 20 de vídeo durante la codificación de los bloques de vídeo para calcular valores interpolados para píxeles subenteros de bloques de referencia. En este caso, la unidad 72 de compensación de movimiento determina los filtros de interpolación usados por el codificador 20 de vídeo a partir de los elementos de sintaxis recibidos y usa los filtros de interpolación para producir bloques predictivos.

Los datos para una imagen de textura correspondiente a un mapa de profundidad se almacenan en la memoria 82 de fotogramas de referencia. La unidad 72 de compensación de movimiento también se configura para interpredicar bloques de profundidad de un mapa de profundidad.

La compresión de imágenes y vídeos ha experimentado un rápido crecimiento, lo que ha llevado a diversos estándares

de codificación. Tales estándares de codificación de vídeo incluyen H.261 de ITU-T, ISO/IEC Grupo de Expertos en Imágenes en Movimiento (MPEG)-1 Parte 2, H.262 de ITU-T u Organización Internacional de Normalización (ISO, por sus siglas en inglés)/Comisión Electrotécnica Internacional (IEC, por sus siglas en inglés) MPEG-2 Parte 2, H.263 de ITU-T, ISO/IEC MPEG-4 Parte 2, Codificación de Vídeo Avanzada (AVC), también conocida como H.264 de ITU-T o ISO/IEC MPEG-4 Parte 10, y Codificación de Vídeo de Alta Eficiencia (HEVC), también conocida como H.265 de ITU-T o MPEG-H Parte 2. La AVC incluye extensiones tales como Codificación de Vídeo Escalable (SVC, por sus siglas en inglés), Codificación de Vídeo Multivista (MVC, por sus siglas en inglés) y Codificación de Vídeo Multivista más Profundidad (MVC+D, por sus siglas en inglés) y AVC 3D (3D-AVC). La HEVC incluye extensiones tales como HEVC Escalable (SHVC, por sus siglas en inglés), HEVC Multivista (MV-HEVC, por sus siglas en inglés) y HEVC 3D (3D-HEVC).

La Codificación de Vídeo Versátil (VVC) es un nuevo estándar de codificación de vídeo, que está siendo desarrollado por el equipo conjunto de expertos en vídeo (JVET, por sus siglas en inglés) de ITU-T e ISO/IEC. En el momento de escribir esta memoria, el último Borrador de Trabajo (WD, por sus siglas en inglés) de VVC está incluido en JVET-K1001-v1. El documento JVET-K0325-v3 de JVET incluye una actualización de la sintaxis de alto nivel de VVC.

En general, la presente divulgación describe técnicas basadas en el subdesarrollo de la norma VVC. Sin embargo, las técnicas también se aplican a otras especificaciones de códecs de vídeo/medios.

Las técnicas de compresión de vídeo realizan predicción espacial (intraimagen) y/o predicción temporal (interimagen) para reducir o eliminar la redundancia inherente en secuencias de vídeo. Para la codificación de vídeo basada en bloques, una porción de vídeo (por ejemplo, una imagen de vídeo o una parte de una imagen de vídeo) se divide en bloques de vídeo, que también se denominan bloques de árbol, bloques de árbol de codificación (CTB, por sus siglas en inglés), unidades de árbol de codificación (CTU, por sus siglas en inglés), unidades de codificación (CU) y/o nodos de codificación. Los bloques de vídeo en una porción intracodificada (I) de una imagen se codifican usando predicción espacial con respecto a muestras de referencia en bloques vecinos en la misma imagen. Los bloques de vídeo en una porción intercodificada (P o B) de una imagen usan predicción espacial con respecto a muestras de referencia en bloques vecinos en la misma imagen o predicción temporal con respecto a muestras de referencia en otras imágenes de referencia. Las imágenes se denominan fotogramas y las imágenes de referencia se denominan fotogramas de referencia.

La predicción espacial o temporal da como resultado un bloque predictivo para un bloque que se ha de codificar. Los datos residuales representan diferencias de píxeles entre el bloque original que se ha de codificar y el bloque predictivo. Un bloque intercodificado se codifica según un vector de movimiento que apunta a un bloque de muestras de referencia que forman el bloque predictivo, y los datos residuales que indican la diferencia entre el bloque codificado y el bloque predictivo. Un bloque intracodificado se codifica según un modo de intracodificación y los datos residuales. Para una mayor compresión, los datos residuales se transforman del dominio de píxeles a un dominio de transformada, dando como resultado coeficientes de transformada residuales, que luego se cuantifican. Los coeficientes de transformada cuantificados, dispuestos inicialmente en una matriz bidimensional, se escanean para producir un vector unidimensional de coeficientes de transformada y se aplica codificación de entropía para lograr una compresión aun mayor.

En una especificación de códec de vídeo, las imágenes se identifican para múltiples propósitos, incluido el uso como imagen de referencia en la interpredicción, para la salida de imágenes del búfer de imágenes decodificadas (DPB, por sus siglas en inglés), para el cambio de escala de vectores de movimiento, para la predicción ponderada, etc. En AVC y HEVC, las imágenes se pueden identificar mediante recuento de orden de imágenes (POC, por sus siglas en inglés). En AVC y HEVC, las imágenes en el DPB se pueden marcar como "utilizadas para referencia a corto plazo", "utilizadas para referencia a largo plazo" o "no utilizadas para referencia". Una vez que una imagen ha sido marcada como "no utilizada para referencia", la imagen ya no se puede usar para la predicción. Cuando la imagen ya no es necesaria para la salida, la imagen se puede eliminar del DPB.

En AVC, hay dos tipos de imágenes de referencia, a corto plazo y a largo plazo. Una imagen de referencia se marca como "no utilizada para referencia" cuando ya no se necesita para la referencia de predicción. La conversión entre estos tres estados (a corto plazo, a largo plazo y no utilizada como referencia) se controla mediante el proceso de marcado de imágenes de referencia decodificadas. Hay dos mecanismos alternativos de marcado de imágenes de referencia decodificadas, el proceso de ventana deslizante implícito y el proceso de operación de control de gestión de memoria (MMCO, por sus siglas en inglés) explícito. El proceso de ventana deslizante marca una imagen de referencia a corto plazo como "no utilizada para referencia" cuando el número de fotogramas de referencia es igual a un número máximo dado (max_num_ref_frames en el conjunto de parámetros de secuencia (SPS, por sus siglas en inglés)). Las imágenes de referencia a corto plazo se almacenan en una forma de 'primera en entrar, primera en salir', de manera que las imágenes a corto plazo decodificadas más recientemente se conservan en el DPB.

El proceso de MMCO explícito incluye múltiples comandos MMCO. Un comando MMCO marca una o más imágenes de referencia a corto o largo plazo como "no utilizadas para referencia", marca todas las imágenes como "no utilizadas para referencia" o marca la imagen de referencia actual o una imagen de referencia a corto plazo existente como a largo plazo y luego asigna un índice de imagen a largo plazo a esa imagen de referencia a largo plazo.

En AVC, las operaciones de marcado de imágenes de referencia, así como los procesos de salida y eliminación de imágenes del DPB, se realizan después de que se haya decodificado una imagen.

5 HEVC introduce un enfoque diferente para la gestión de imágenes de referencia, denominado conjunto de imágenes de referencia (RPS, por sus siglas en inglés). La diferencia más fundamental con el concepto RPS en comparación con el proceso de MMCO/ventana deslizante de AVC es que para cada porción en particular se proporciona un conjunto completo de las imágenes de referencia utilizadas por la imagen actual o cualquier imagen posterior. Por lo tanto, se señala un conjunto completo de todas las imágenes que se deben mantener en el DPB para su uso por la imagen actual o futura. Esto es diferente del esquema AVC donde sólo se señalizan cambios relativos en el DPB. Con el concepto RPS, no se necesita información de imágenes anteriores en orden de decodificación para mantener el estado correcto de las imágenes de referencia en el DPB.

15 En HEVC, el orden de decodificación de imágenes y las operaciones de DPB están modificados en comparación con AVC para aprovechar las ventajas del RPS y mejorar la resistencia a los errores. En AVC, las operaciones de marcado de imágenes y de búfer (tanto la salida como la eliminación de imágenes decodificadas del DPB) generalmente se aplican después de que se haya decodificado una imagen actual. En HEVC, el RPS se decodifica primero a partir de un encabezado de porción de la imagen actual, luego se aplican generalmente operaciones de marcado de imágenes y de búfer antes de decodificar la imagen actual.

20 Cada encabezado de porción en HEVC debe incluir parámetros para la señalización del RPS para la imagen que contiene las porciones. La única excepción es que no se señala ningún RPS para las porciones de Refresco de Decodificación Instantánea (IDR, por sus siglas en inglés). En cambio, se deduce que el RPS está vacío. Para las porciones I que no pertenecen a una imagen IDR, se proporciona un RPS, incluso si pertenecen a una imagen I, ya que hay imágenes que siguen a la imagen I en orden de decodificación que utilizan interpredicción de imágenes que precedían a la imagen I en orden de decodificación. El número de imágenes en un RPS no deberá exceder el límite de tamaño del DPB especificado por el elemento de sintaxis `sps_max_dec_pic_buffering` en el SPS.

30 Cada imagen está asociada con un valor POC que representa el orden de salida. Los encabezados de porción contienen una palabra código de longitud fija, `pic_order_cnt_lsb`, que representa los bits menos significativos (LSB, por sus siglas en inglés) del valor POC completo, también conocidos como LSB de POC. La longitud de la palabra código se señala en el SPS y puede ser de entre 4 y 16 bits. El concepto RPS utiliza POC para identificar imágenes de referencia. Además de su propio valor POC, cada encabezado de porción contiene directamente o hereda del SPS una representación codificada de los valores POC (o los LSB) de cada imagen en el RPS.

35 El RPS para cada imagen consta de cinco listas diferentes de imágenes de referencia, también denominadas los cinco subconjuntos de RPS. `RefPicSetStCurrBefore` consiste en todas las imágenes de referencia a corto plazo que son anteriores a la imagen actual, en orden tanto de decodificación como de salida, y que se utilizan en la interpredicción de la imagen actual. `RefPicSetStCurrAfter` consiste en todas las imágenes de referencia a corto plazo que son anteriores a la imagen actual en orden de decodificación, que suceden a la imagen actual en orden de salida, y que se utilizan en la interpredicción de la imagen actual. `RefPicSetStFoll` consiste en todas las imágenes de referencia a corto plazo que se utilizan en la interpredicción de una o más de las imágenes que siguen a la imagen actual en orden de decodificación y que no se utilizan en la interpredicción de la imagen actual. `RefPicSetLtCurr` consiste en todas las imágenes de referencia a largo plazo que se utilizan en la interpredicción de la imagen actual. `RefPicSetLtFoll` consiste en todas las imágenes de referencia a largo plazo que se utilizan en la interpredicción de una o más de las imágenes que siguen a la imagen actual en orden de decodificación y que no se utilizan en la interpredicción de la imagen actual.

50 El RPS se señala utilizando hasta tres bucles que iteran diferentes tipos de imágenes de referencia; imágenes de referencia a corto plazo con un valor POC más bajo que la imagen actual, imágenes de referencia a corto plazo con un valor POC más alto que la imagen actual, e imágenes de referencia a largo plazo. Además, se envía un señalizador (`used_by_curr_pic_X_flag`) para cada imagen de referencia que indica si la imagen de referencia es utilizada para referencia por la imagen actual (incluida en una de las listas `RefPicSetStCurrBefore`, `RefPicSetStCurrAfter` o `RefPicSetLtCurr`) o no (incluida en una de las listas `RefPicSetStFoll` o `RefPicSetLtFoll`).

55 La Figura 4 ilustra un RPS 400 que tiene una imagen actual B14 con entradas (por ejemplo, una imagen) en todos los subconjuntos 402 del RPS 400. En el ejemplo de la Figura 4, la imagen actual B14 contiene exactamente una imagen en cada uno de los cinco subconjuntos 402 (también conocidos como subconjuntos RPS). P8 es la imagen del subconjunto 402 denominada `RefPicSetStCurrBefore` porque la imagen está antes en orden de salida y la utiliza B14. P12 es la imagen del subconjunto 402 denominada `RefPicSetStCurrAfter` porque la imagen está después en orden de salida y la utiliza B14. P13 es la imagen del subconjunto 402 denominada `RefPicSetStFoll` porque la imagen es una imagen de referencia a corto plazo no utilizada por B14 (pero debe mantenerse en el DPB ya que la utiliza B15). P4 es la imagen del subconjunto 402 denominada `RefPicSetLtCurr` porque la imagen es una imagen de referencia a largo plazo utilizada por B14. I0 es la imagen del subconjunto 402 denominada `RefPicSetLtFoll`, ya que la imagen es una imagen de referencia a largo plazo no utilizada por la imagen actual (pero debe mantenerse en el DPB, ya que la utiliza B15).

La parte a corto plazo del RPS 400 se incluye directamente en el encabezado de porción. Como alternativa, el encabezado de porción contiene sólo un elemento de sintaxis que representa un índice, haciendo referencia a una lista predefinida de RPS enviados en el SPS activo. La parte a corto plazo del RPS 402 se puede señalar utilizando cualquiera de dos esquemas diferentes; Inter RPS, como se describe posteriormente, o Intra RPS, como se describe aquí. Cuando se utiliza Intra RPS, num_negative_pics y num_positive_pics se señalan representando la longitud de dos listas diferentes de imágenes de referencia. Estas listas contienen las imágenes de referencia con diferencia de POC negativa y diferencia de POC positiva en comparación con la imagen actual, respectivamente. Cada elemento de estas listas está codificado con un código de longitud variable que representa la diferencia en el valor POC en relación con el elemento anterior de la lista menos uno. Para la primera imagen de cada lista, la señalización es relativa al valor POC de la imagen actual menos uno.

Al codificar los RPS recurrentes en el conjunto de parámetros de secuencia, es posible codificar los elementos de un RPS (por ejemplo, RPS 400) con referencia a otro RPS ya codificado en el conjunto de parámetros de secuencia. Esto se denomina Inter RPS. No hay problemas de robustez contra errores asociados con este método, ya que todos los RPS del conjunto de parámetros de secuencia están en la misma unidad de capa de abstracción de red (NAL, por sus siglas en inglés). La sintaxis Inter RPS aprovecha el hecho de que el RPS de la imagen actual se puede predecir a partir del RPS de una imagen previamente decodificada. Esto se debe a que todas las imágenes de referencia de la imagen actual deben ser, bien imágenes de referencia de la imagen anterior, bien la propia imagen decodificada anteriormente. Sólo es necesario indicar cuáles de estas imágenes deben ser imágenes de referencia y utilizarse para la predicción de la imagen actual. Por lo tanto, la sintaxis comprende lo siguiente: un índice que apunta al RPS para su uso como predictor, un delta_POC que se ha de agregar al delta_POC del predictor para obtener el delta POC del RPS actual, y un conjunto de indicadores para indicar qué imágenes son imágenes de referencia y si sólo se usan para la predicción de imágenes futuras.

Los codificadores que deseen aprovechar el uso de imágenes de referencia a largo plazo deben establecer el elemento de sintaxis SPS long_term_ref_pics_present_flag en uno. Las imágenes de referencia a largo plazo se pueden señalar entonces en el encabezado de porción mediante palabras código de longitud fija, poc_lsb_lt, que representan los bits menos significativos del valor POC completo de cada imagen a largo plazo. Cada poc_lsb_lt es una copia de la palabra código pic_order_cnt_lsb que se señaló para una imagen a largo plazo en particular. También es posible señalar un conjunto de imágenes a largo plazo en el SPS como una lista de valores LSB de POC. Los LSB de POC para una imagen a largo plazo pueden señalizarse entonces en el encabezado de porción como un índice para esta lista.

El elemento de sintaxis delta_poc_msb_cycle_lt_minus1 se puede señalar adicionalmente para posibilitar el cálculo de la distancia POC completa de una imagen de referencia a largo plazo en relación con la imagen actual. Se requiere que la palabra código delta_poc_msb_cycle_lt_minus1 esté señalizada para cada imagen de referencia a largo plazo que tenga el mismo valor LSB de POC que cualquier otra imagen de referencia en el RPS.

Para el marcado de imágenes de referencia en HEVC, normalmente habrá varias imágenes presentes en el DPB antes de la decodificación de imágenes. Algunas de las imágenes están disponibles para la predicción y, como tales, marcadas como "utilizadas para referencia". Otras imágenes no están disponibles para la predicción, pero están esperando la salida y, como tales, marcadas como "no utilizadas para referencia". Cuando se ha analizado el encabezado de porción, se lleva a cabo un proceso de marcado de imágenes antes de decodificar los datos de la porción. Las imágenes que están presentes en el DPB y marcadas como "utilizadas para referencia", pero que no están incluidas en el RPS, están marcadas como "no utilizadas para referencia". Las imágenes que no están presentes en el DPB, pero que se incluyen en el conjunto de imágenes de referencia, se ignoran cuando used_by_curr_pic_X_flag es igual a cero. Sin embargo, cuando used_by_curr_pic_X_flag es en su lugar igual a uno, esta imagen de referencia estaba destinada a ser utilizada para la predicción en la imagen actual, pero falta. Entonces se infiere una pérdida de imagen no intencionada y el decodificador debe tomar las medidas apropiadas.

Después de decodificar la imagen actual, se marca como "utilizada para referencia a corto plazo".

A continuación, se expone la construcción de la lista de imágenes de referencia en HEVC. En HEVC, el término interpredicción se utiliza para denotar la predicción obtenida de elementos de datos (por ejemplo, valores de muestra o vectores de movimiento) de imágenes de referencia distintas de la imagen decodificada actual. Al igual que en AVC, una imagen se puede predecir a partir de múltiples imágenes de referencia. Las imágenes de referencia que se utilizan para la interpredicción se organizan en una o más listas de imágenes de referencia. El índice de referencia identifica cuál de las imágenes de referencia de la lista debe utilizarse para crear la señal de predicción.

Se utilizan una sola lista de imágenes de referencia, Lista 0, para una porción P y dos listas de imágenes de referencia, Lista 0 y Lista 1, para las porciones B. De manera similar a AVC, la construcción de la lista de imágenes de referencia en HEVC incluye la inicialización de la lista de imágenes de referencia y la modificación de la lista de imágenes de referencia.

En AVC, el proceso de inicialización para la Lista 0 es diferente para la porción P (para la que se utiliza el orden de decodificación) y las porciones B (para las que se utiliza el orden de salida). En HEVC, el orden de salida se utiliza en

ambos casos.

La inicialización de la lista de imágenes de referencia crea la Lista 0 y la Lista 1 por defecto (si la porción es una porción B) basándose en tres subconjuntos RPS: RefPicSetStCurrBefore, RefPicSetStCurrAfter y RefPicSetLtCurr. Las imágenes a corto plazo con un orden de salida anterior (posterior) se insertan primero en la Lista 0 (Lista 1) en orden ascendente de distancia POC a la imagen actual, luego las imágenes a corto plazo con un orden de salida posterior (anterior) se insertan en la Lista 0 (Lista 1) en orden ascendente de distancia POC a la imagen actual, y luego, finalmente, las imágenes a largo plazo se insertan al final. En términos de RPS, para la Lista 0, las entradas de RefPicSetStCurrBefore se insertan en la lista inicial, seguidas de las entradas de RefPicSetStCurrAfter. Posteriormente, se anexan las entradas de RefPicSetLtCurr, si están disponibles.

En HEVC, se repite el proceso anterior (se vuelven a agregar las imágenes de referencia que ya se han agregado a la lista de imágenes de referencia) cuando el número de entradas de una lista es menor que el número objetivo de imágenes de referencia activas (señalizadas en el conjunto de parámetros de imagen o en el encabezado de porción). Cuando el número de entradas es mayor que el número objetivo, la lista se trunca.

Una vez inicializada una lista de imágenes de referencia, se modifica de manera que las imágenes de referencia para la imagen actual se organicen en cualquier orden, incluido el caso en que una imagen de referencia en particular aparezca en más de una posición de la lista, en función de los comandos de modificación de la lista de imágenes de referencia. Cuando el señalizador que indica la presencia de modificaciones de la lista se establece en uno, se señalizan un número fijo (igual al número objetivo de entradas en la lista de imágenes de referencia) de comandos y cada comando inserta una entrada para una lista de imágenes de referencia. Una imagen de referencia se identifica en el comando por el índice para la lista de imágenes de referencia para la imagen actual obtenida a partir de la señalización RPS. Esto es diferente de la modificación de la lista de imágenes de referencia en H.264/AVC, donde una imagen se identifica ya sea por el número de imagen (obtenida a partir del elemento de sintaxis frame_num) o el índice de imagen de referencia a largo plazo, y es posible que se necesiten menos comandos, por ejemplo, para intercambiar las dos primeras entradas de una lista inicial o insertar una entrada al comienzo de la lista inicial y cambiar las demás.

No se permite que una lista de imágenes de referencia incluya ninguna imagen de referencia con un TemporalId mayor que la imagen actual. Un flujo de bits HEVC puede consistir en varias subcapas temporales. Cada unidad NAL pertenece a una subcapa específica según lo indicado por TemporalId (igual a temporal_id_plus1 - 1).

La gestión de imágenes de referencia se basa directamente en listas de imágenes de referencia. El documento JCTVC-G643 de JCT-VC incluye un enfoque para utilizar directamente tres listas de imágenes de referencia, lista 0 de imágenes de referencia, lista 1 de imágenes de referencia y una lista de imágenes de referencia libre, para la gestión de las imágenes de referencia en el DPB, evitando así la necesidad de los procesos de señalización y decodificación, incluidos, bien 1) los procesos de ventana deslizante y de MMCO, así como los procesos de inicialización y modificación de la lista de imágenes de referencia en AVC, bien 2) el conjunto de imágenes de referencia, así como los procesos de inicialización y modificación de la lista de imágenes de referencia en HEVC.

Los enfoques para la gestión de imágenes de referencia tienen varios problemas. El enfoque AVC involucra la ventana deslizante, los procesos MMCO y los procesos de inicialización y modificación de la lista de imágenes de referencia, que son complejos. Además, la pérdida de imágenes conduce a la pérdida del estado del DPB en términos de qué imágenes deberían haber estado en el DPB para fines de referencia de interpredicción posteriores. El enfoque HEVC no tiene el problema de pérdida de estado del DPB. Sin embargo, el enfoque HEVC implica un complejo proceso de señalización y obtención de conjuntos de imágenes de referencia, así como los procesos de inicialización y modificación de listas de imágenes de referencia, que son complejos. El enfoque en JCTVC-G643 para utilizar directamente tres listas de imágenes de referencia, lista 0 de imágenes de referencia, lista 1 de imágenes de referencia, así como una lista de imágenes de referencia libre, para la gestión de las imágenes de referencia en el DPB implica los siguientes aspectos: una tercera lista de imágenes de referencia, es decir, la lista de imágenes de referencia libre; la codificación en dos partes de las diferencias de POC como una parte "a corto plazo" y una parte "a largo plazo" codificada por ue(v); la granularidad de POC basada en TemporalId para la codificación de diferencias de POC, el uso de la codificación en dos partes de las diferencias de POC para determinar el marcado entre "utilizada para referencia a corto plazo" o "utilizada para referencia a largo plazo"; una descripción del subconjunto de listas de imágenes de referencia que posibilite la capacidad de especificar una lista de imágenes de referencia mediante la eliminación de imágenes de referencia de la cola de una determinada descripción de lista de imágenes de referencia anterior; el modo de copia de listas de imágenes de referencia posibilitado por el elemento de sintaxis ref_pic_list_copy_flag; y el proceso de descripción de listas de imágenes de referencia. Cada uno de los aspectos anteriores hace que el enfoque sea innecesariamente complejo. Además, el proceso de decodificación para listas de imágenes de referencia en JCTVC-G643 también es complejo. La señalización de imágenes de referencia a largo plazo necesita la señalización del ciclo POC en los encabezados de porción. Esto no es eficaz.

Con el fin de abordar los problemas enumerados anteriormente, en la presente memoria se divulgan las siguientes soluciones, cada una de las cuales puede aplicarse individualmente y algunas de las cuales pueden aplicarse en combinación. 1) El marcado de la imagen de referencia se basa directamente en las dos listas de imágenes de

referencia, a saber, la lista 0 de imágenes de referencia y la lista 1 de imágenes de referencia. 1a) La información para la obtención de las dos listas de imágenes de referencia se señala en función de elementos de sintaxis y estructuras de sintaxis en el SPS, PPS y/o el encabezado de porción. 1b) Cada una de las dos listas de imágenes de referencia para una imagen se señala explícitamente en una estructura de lista de imágenes de referencia. 1b.i) Una o más estructuras de lista de imágenes de referencia pueden señalizarse en SPS y cada una de ellas puede ser referenciada por un índice desde el encabezado de porción. 1b.ii) Cada una de las listas 0 y 1 de imágenes de referencia se puede señalar directamente en el encabezado de porción. 2) La información para la obtención de las dos listas de imágenes de referencia está señalizada para todos los tipos de porciones, es decir, porciones B (bipredictiva), P (unipredictiva) e I (intra). El término porción se refiere a una colección de unidades de árbol de codificación, tales como una porción en HEVC o el último WD de VVC; también se refiere a alguna otra colección de unidades de árbol de codificación, tales como un mosaico (*tile*) en HEVC. 3) Las dos listas de imágenes de referencia se generan para todos los tipos de porciones, es decir, porciones B, P e I. 4) Las dos listas de imágenes de referencia se construyen directamente sin utilizar un proceso de inicialización de lista de imágenes de referencia y un proceso de modificación de lista de imágenes de referencia. 5) En cada una de las dos listas de imágenes de referencia, las imágenes de referencia que se utilizan para la interpolación de la imagen actual sólo pueden ser referenciadas por una serie de entradas al principio de la lista. Estas entradas se denominan entradas activas en la lista, mientras que otras entradas se denominan entradas inactivas en la lista. El número de entradas totales y el número de entradas activas en la lista pueden obtenerse ambos. 6) No se permite que a la imagen a la que hace referencia una entrada inactiva en una lista de imágenes de referencia haga referencia otra entrada en la lista de imágenes de referencia ni cualquier entrada en la otra lista de imágenes de referencia. 7) Las imágenes de referencia a largo plazo sólo se identifican mediante cierto número de LSB de POC, donde este número es mayor que el número de LSB de POC señalizados en los encabezados de porción para la obtención de valores POC, y este número se indica en el SPS. 8) Las estructuras de lista de imágenes de referencia se señalan sólo en encabezados de porción, tanto las imágenes de referencia a corto plazo como las imágenes de referencia a largo plazo se identifican mediante sus LSB de POC, que están representados por números de bits que son diferentes del número de bits utilizado para representar los LSB de POC señalizados en encabezados de porción para la obtención de valores POC, y el número de bits utilizado para representar los LSB de POC para identificar imágenes de referencia a corto plazo e imágenes de referencia a largo plazo es diferente. 9) Las estructuras de lista de imágenes de referencia se señalan sólo en encabezados de porción, no se hace distinción entre imágenes de referencia a corto y largo plazo, todas las imágenes de referencia se denominan simplemente imágenes de referencia y las imágenes de referencia se identifican por sus LSB de POC, que están representados por un número de bits que es diferente del número de bits utilizado para representar los LSB de POC señalizados en encabezados de porción para la obtención de valores POC.

Se proporciona una primera realización de la presente divulgación. La descripción es relativa al último WD de VVC. En esta realización, se señalan en el SPS dos conjuntos de estructuras de lista de imágenes de referencia, uno para la lista 0 de imágenes de referencia y otro para la lista 1 de imágenes de referencia.

Se proporcionan definiciones de algunas de las expresiones utilizadas en la presente memoria. Una imagen de punto de acceso intraaleatorio (IRAP): una imagen codificada para la cual cada unidad NAL de capa de codificación de vídeo (VCL, por sus siglas en inglés) tiene `nal_unit_type` igual a `IRAP_NUT`. Imagen no-IRAP: una imagen codificada para la cual cada unidad NAL de VCL tiene `nal_unit_type` igual a `NON_IRAP_NUT`. Lista de imágenes de referencia: una lista de imágenes de referencia que se utiliza para la interpolación de una porción P o B. Se generan dos listas de imágenes de referencia, la lista 0 de imágenes de referencia y la lista 1 de imágenes de referencia, para cada porción de una imagen no-IRAP. El conjunto de imágenes únicas a las que hacen referencia todas las entradas de las dos listas de imágenes de referencia asociadas con una imagen consiste en todas las imágenes de referencia que se utilizan para la interpolación de la imagen asociada o cualquier imagen que siga a la imagen asociada en orden de decodificación. Para decodificar los datos de una porción P, sólo se utiliza la lista 0 de imágenes de referencia para la interpolación. Para decodificar los datos de una porción B, se utilizan ambas listas de imágenes de referencia para la interpolación. Para decodificar los datos de una porción I, no se utiliza ninguna lista de imágenes de referencia para la interpolación. Imagen de referencia a largo plazo (LTRP, por sus siglas en inglés): una imagen que se marca como "utilizada para referencia a largo plazo". Imagen de referencia a corto plazo (STRP, por sus siglas en inglés): una imagen que se marca como "utilizada para referencia a corto plazo".

Las expresiones "utilizada para referencia a corto plazo", "utilizada para referencia a largo plazo" o "no utilizada para referencia" se definen en VVC en la sección 8.3.3 Proceso de decodificación para el marcado de imágenes de referencia, se definen en HEVC en la sección 8.3.2 Proceso de decodificación para el conjunto de imágenes de referencia y se definen en AVC en la sección 7.4.3.3 Semántica de marcado de imágenes de referencia decodificadas. Tal como se utilizan en la presente memoria, las expresiones tienen el mismo significado.

A continuación se proporcionan la sintaxis y la semántica pertinentes para la primera realización.

Sintaxis de encabezado de unidad NAL.

<code>nal_unit_header() {</code>	Descriptor
<code>forbidden_zero_bit</code>	<code>f(1)</code>

ES 3 014 003 T3

nal_unit_type	u(5)
nuh_temporal_id_plus1	u(3)
nuh_reserved_zero_7bits	u(7)
}	

Sintaxis de Carga Útil de Secuencia de Bytes sin Procesar (RBSP, por sus siglas en inglés) de conjunto de parámetros de secuencia.

seq_parameter_set_rbsp() {	Descriptor
sps_seq_parameter_set_id	ue(v)
chroma_format_idc	ue(v)
if(chroma_format_idc == 3)	
separate_colour_plane_flag	u(1)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
bit_depth_luma_minus8	ue(v)
bit_depth_chroma_minus8	ue(v)
log2_max_pic_order_cnt_lsb_minus4	ue(v)
sps_max_dec_pic_buffering_minus1	ue(v)
qtbtt_dual_tree_intra_flag	ue(v)
log2_ctu_size_minus2	ue(v)
log2_min_qt_size_intra_slices_minus2	ue(v)
log2_min_qt_size_inter_slices_minus2	ue(v)
max_mtt_hierarchy_depth_inter_slices	ue(v)
max_mtt_hierarchy_depth_intra_slices	ue(v)
long_term_ref_pics_flag	u(1)
if(long_term_ref_pics_flag)	
additional_lt_poc_lsb	ue(v)
for(i = 0; i < 2; i++) {	
num_ref_pic_lists_in_sps[i]	ue(v)
for(j = 0; j < num_ref_pic_lists_in_sps[i]; j++)	
ref_pic_list_struct(i, j, long_term_ref_pics_flag)	
}	
rbsp_trailing_bits()	
}	

5

Sintaxis de RBSP de conjunto de parámetros de imagen.

pic_parameter_set_rbsp() {	Descriptor
pps_pic_parameter_set_id	ue(v)
pps_seq_parameter_set_id	ue(v)
for(i = 0; i < 2; i++)	
num_ref_idx_default_active_minus1[i]	ue(v)
rbsp_trailing_bits()	
}	

Sintaxis de encabezado de porción.

10

slice_header() {	Descriptor
slice_pic_parameter_set_id	ue(v)
slice_address	u(v)
slice_type	ue(v)
if (slice_type != I)	
log2_diff_ctu_max_bt_size	ue(v)

if(nal_unit_type != IRAP_NUT) {	
slice_pic_order_cnt_lsb	u(v)
for(i = 0; i < 2; i++) {	
ref_pic_list_sps_flag[i]	u(1)
if(ref_pic_list_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1)	
ref_pic_list_idx[i]	u(v)
} else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i], long_term_ref_pics_flag)	
}	
if(slice_type == P slice_type == B) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag)	
for(i = 0; i < (slice_type == B ? 2 : 1); i++)	
num_ref_idx_active_minus1[i]	ue(v)
}	
}	
byte_alignment()	
}	

Sintaxis de estructura de lista de imágenes de referencia.

ref_pic_list_struct(listIdx, rplIdx, ltrpFlag) {	Descriptor
num_strp_entries[listIdx][rplIdx]	ue(v)
if(ltrpFlag)	
num_ltrp_entries[listIdx][rplIdx]	ue(v)
for(i = 0; i < NumEntriesInList[listIdx][rplIdx]; i++) {	
if(num_ltrp_entries[listIdx][rplIdx] > 0)	
lt_ref_pic_flag[listIdx][rplIdx][i]	
if(!lt_ref_pic_flag[listIdx][rplIdx][i])	
delta_poc_st [listIdx][rplIdx][i] // No utilizándose "delta_poc_st_minus1" aquí, para permitir que la misma imagen de referencia se incluya en múltiples posiciones de una lista de imágenes de referencia. Como efecto secundario, esto también posibilitaría el referenciado de la imagen actual (es decir, copia intrabloque). Cabe señalar también que está firmado.	se(v)
else	
poc_lsb_lt[listIdx][rplIdx][i]	u(v)
}	
}	

5 Semántica de encabezado de unidad NAL.

Un forbidden_zero_bit será igual a 0. nal_unit_type especifica el tipo de estructura de datos RBSP contenida en la unidad NAL.

10 Tabla 7-1 - Códigos de tipo de unidad NAL y clases de tipo de unidad NAL

nal_unit_type	Nombre de nal_unit_type	Contenido de unidad NAL y estructura de sintaxis de RBSP	Clase de tipo de unidad NAL
0	NON_IRAP_NUT	Segmento de porción codificada de una imagen no-IRAP slice_layer_rbsp()	VCL
1	IRAP_NUT	Porción codificada de una imagen IRAP slice_layer_rbsp()	VCL
2-15	RSV_VCL_NUT	Unidades NAL de VCL reservadas	VCL
16	SPS_NUT	Conjunto de parámetros de secuencia seq_parameter_set_rbsp()	no-VCL
17	PPS_NUT	Conjunto de parámetros de imagen	no-VCL

		pic_parameter_set_rbsp()	
18	EOS_NUT	Fin de secuencia end_of_seq_rbsp()	no-VCL
19	EOB_NUT	Fin de flujo de bits end_of_bitstream_rbsp()	no-VCL
20, 21	PREFIX_SEI_NUT SUFFIX_SEI_NUT	Información de mejora suplementaria sei_rbsp()	no-VCL
22-26	RSV_NVCL	Reservados	no-VCL
27-31	UNSPEC	Sin especificar	no-VCL

El nuh_temporal_id_plus1 minus 1 especifica un identificador temporal para la unidad NAL. El valor de nuh_temporal_id_plus1 no será igual a 0. La variable TemporalId se especifica como sigue: TemporalId = nuh_temporal_id_plus1 - 1. Cuando nal_unit_type es igual a IRAP_NUT, la porción codificada pertenece a una imagen IRAP, TemporalId será igual a 0. El valor de TemporalId será el mismo para todas las unidades NAL de VCL de una unidad de acceso. El valor de TemporalId de una imagen codificada o de una unidad de acceso es el valor de TemporalId de las unidades NAL de VCL de la imagen codificada o de la unidad de acceso. El valor de TemporalId para las unidades NAL no-VCL se limita de la siguiente manera: si nal_unit_type es igual a SPS_NUT, TemporalId será igual a 0 y TemporalId de la unidad de acceso que contiene la unidad NAL será igual a 0. De lo contrario, si nal_unit_type es igual a EOS_NUT o EOB_NUT, TemporalId será igual a 0. De lo contrario, TemporalId será mayor o igual que TemporalId de la unidad de acceso que contiene la unidad NAL. Cuando la unidad NAL es una unidad NAL no-VCL, el valor de TemporalId es igual al valor mínimo de los valores de TemporalId de todas las unidades de acceso a las que se aplica la unidad NAL no-VCL. Cuando nal_unit_type es igual a PPS_NUT, TemporalId es mayor o igual que TemporalId de la unidad de acceso continente, ya que todos los conjuntos de parámetros de imagen (PPS, por sus siglas en inglés) se incluyen en el comienzo de un flujo de bits, en donde la primera imagen codificada tiene TemporalId igual a 0. Cuando nal_unit_type es igual a PREFIX_SEI_NUT o SUFFIX_SEI_NUT, TemporalId es mayor o igual que TemporalId de la unidad de acceso continente, ya que una unidad NAL de SEI contiene información que se aplica a un subconjunto de flujo de bits que incluye unidades de acceso para las que los valores de TemporalId son mayores que TemporalId de la unidad de acceso que contiene la unidad NAL de SEI. nuh_reserved_zero_7bits será igual a '0000000'. Otros valores de nuh_reserved_zero_7bits serán especificados en el futuro por ITU-T | ISO/IEC. Los decodificadores deberán ignorar (es decir, eliminar del flujo de bits y descartar) las unidades NAL con valores de nuh_reserved_zero_7bits no iguales a '0000000'.

Semántica de RBSP de conjunto de parámetros de secuencia.

Un log2_max_pic_order_cnt_lsb_minus4 especifica el valor de la variable MaxPicOrderCntLsb que se utiliza en el proceso de decodificación para el recuento de orden de imágenes de la siguiente manera: MaxPicOrderCntLsb = 2 (log2_max_pic_order_cnt_lsb_minus4 + 4). El valor de log2_max_pic_order_cnt_lsb_minus4 estará en el intervalo de 0 a 12, inclusive. sps_max_dec_pic_buffering_minus1 plus 1 especifica el tamaño máximo requerido del búfer de imágenes decodificadas para la CVS en unidades de búferes de almacenamiento de imágenes. El valor de sps_max_dec_pic_buffering_minus1 estará en el intervalo de 0 a MaxDpbSize - 1, inclusive, donde MaxDpbSize es como se especifica en otro lugar. long_term_ref_pics_flag igual a 0 especifica que no se usa LTRP para la interpredicción de ninguna imagen codificada en la CVS. long_term_ref_pics_flag igual a 1 especifica que se usan LTRP para la interpredicción de una o más imágenes codificadas en la CVS. additional_lt_poc_lsb especifica el valor de la variable MaxLtPicOrderCntLsb que se usa en el proceso de decodificación para las listas de imágenes de referencia de la siguiente manera: MaxLtPicOrderCntLsb = 2(log2_max_pic_order_cnt_lsb_minus4 + 4 + additional_lt_poc_lsb). El valor de additional_lt_poc_lsb estará en el intervalo de 0 a 32 - log2_max_pic_order_cnt_lsb_minus4 - 4, inclusive. Cuando no está presente, el valor de additional_lt_poc_lsb se deduce que es igual a 0. num_ref_pic_lists_in_sps[i] especifica el número de las estructuras de sintaxis ref_pic_list_struct(listIdx, rplIdx, ltrpFlag) con listIdx igual a i incluidas en el SPS. El valor de num_ref_pic_lists_in_sps[i] estará en el intervalo de 0 a 64, inclusive. Para cada valor de listIdx (igual a 0 o 1), un decodificador debe asignar memoria para un número total de estructuras de sintaxis num_ref_pic_lists_in_sps[i] + 1 ref_pic_list_struct(listIdx, rplIdx, ltrpFlag), ya que hay una estructura de sintaxis ref_pic_list_struct(listIdx, rplIdx, ltrpFlag) señalizada directamente en los encabezados de porción de una imagen actual.

Semántica de RBSP de conjunto de parámetros de imagen.

Un num_ref_idx_default_active_minus1 [i] plus 1, cuando i es igual a 0, especifica el valor deducido de la variable NumRefIdxActive[0] para las porciones P o B con num_ref_idx_active_override_flag igual a 0 y, cuando i es igual a 1, especifica el valor deducido de NumRefIdxActive[1] para las porciones B con num_ref_idx_active_override_flag igual a 0. El valor de num_ref_idx_default_active_minus1[i] estará en el intervalo de 0 a 14, inclusive.

Semántica de encabezado de porción.

Cuando estén presentes, el valor de cada uno de los elementos de sintaxis de encabezado de porción slice_pic_parameter_set_id y slice_pic_order_cnt_lsb será el mismo en todos los encabezados de porción de una imagen codificada.... slice_type especifica el tipo de codificación de la porción de acuerdo con la Tabla 7-3.

Tabla 7-3 - Asociación de nombres a slice_type

slice_type	Nombre de slice_type
0	B (porción B)
1	P (porción P)
2	I (porción I)

- 5 Cuando nal_unit_type es igual a IRAP_NUT, es decir, la imagen es una imagen IRAP, slice_type será igual a 2. ...
- slice_pic_order_cnt_lsb especifica el módulo de recuento de orden de imágenes MaxPicOrderCntLsb para la imagen actual. La longitud del elemento de sintaxis slice_pic_order_cnt_lsb es $\log_2 \text{max_pic_order_cnt_lsb_minus4} + 4$ bits. El valor de slice_pic_order_cnt_lsb estará en el intervalo de 0 a MaxPicOrderCntLsb - 1, inclusive. Cuando slice_pic_order_cnt_lsb no está presente, se deduce que slice_pic_order_cnt_lsb es igual a 0. ref_pic_list_sps_flag[i] igual a 1 especifica que la lista i de imágenes de referencia de la imagen actual se obtiene sobre la base de una de las estructuras de sintaxis ref_piclist_struct(listldx, rplldx, ltrpFlag) con listldx igual a i en el SPS activo.
- 10 ref_pic_list_sps_flag[i] igual a 0 especifica que la lista i de imágenes de referencia de la imagen actual se obtiene sobre la base de la estructura de sintaxis ref_pic_list_struct(listldx, rplldx, ltrpFlag) con listldx igual a i que se incluye directamente en los encabezados de porción de la imagen actual. Cuando num_ref_pic_lists_in_sps[i] es igual a 0, el valor de ref_pic_list_sps_flag[i] será igual a 0. ref_pic_list_idx[i] especifica el índice, en la lista de la estructura de sintaxis ref_pic_list_struct(listldx, rplldx, ltrpFlag) con listldx igual a i que se incluye en el SPS activo, de la estructura de sintaxis ref_pic_list_struct(listldx, rplldx, ltrpFlag) con listldx igual a i que se utiliza para la obtención de la lista i de imágenes de referencia de la imagen actual. El elemento de sintaxis ref_pic_list_idx[i] está representado por $\text{Ceil}(\log_2(\text{num_ref_pic_lists_in_sps}[i]))$ bits. Cuando no está presente, el valor de ref_pic_list_idx[i] se deduce que es igual a 0. El valor de ref_pic_list_idx[i] debe estar en el intervalo de 0 a num_ref_pic_lists_in_sps[i] - 1, inclusive.
- 20 num_ref_idx_active_override_flag igual a 1 especifica que el elemento de sintaxis num_ref_idx_active_minus1[0] está presente para porciones P y B y que el elemento de sintaxis num_ref_idx_active_minus1[1] está presente para porciones B. num_ref_idx_active_override_flag igual a 0 especifica que los elementos de sintaxis num_ref_idx_active_minus1[0] y num_ref_idx_active_minus1[1] no están presentes. num_ref_idx_active_minus1[i], cuando está presente, especifica el valor de la variable NumRefIdxActive[i] de la siguiente manera:
- 25 NumRefIdxActive[i] = num_ref_idx_active_minus1[i] + 1. El valor de num_ref_idx_active_minus1[i] estará en el intervalo de 0 a 14, inclusive.

- El valor de NumRefIdxActive[i] - 1 especifica el índice de referencia máximo para la lista i de imágenes de referencia que se utiliza para decodificar la porción. Cuando el valor de NumRefIdxActive[i] es igual a 0, no se utiliza ningún índice de referencia para la lista i de imágenes de referencia para decodificar la porción. Para i igual a 0 o 1, cuando la porción actual es una porción B y num_ref_idx_active_override_flag es igual a 0, se deduce que NumRefIdxActive[i] es igual a num_ref_idx_default_active_minus1[i] + 1. Cuando la porción actual es una porción P y num_ref_idx_active_override_flag es igual a 0, se deduce que NumRefIdxActive[0] es igual a num_ref_idx_default_active_minus1[0] + 1. Cuando la porción actual es una porción P, se deduce que NumRefIdxActive[1] es igual a 0. Cuando la porción actual es una porción I, se deduce que tanto NumRefIdxActive[0] como NumRefIdxActive[1] son iguales a 0.
- 30
- 35

- Como alternativa, para i igual a 0 o 1, se aplica lo siguiente después de lo anterior: sea rplldx1 establecido igual a ref_pic_list_sps_flag[i] ? ref_pic_list_idx[i] : num_ref_pic_lists_in_sps[i], y numRpEntries[i] igual a num_strp_entries[i][rplldx1] + num_ltrp_entries[i][rplldx1]. Cuando NumRefIdxActive[i] es mayor que numRpEntries[i], el valor de NumRefIdxActive[i] se establece igual a numRpEntries[i]
- 40

Semántica de estructura de lista de imágenes de referencia.

- 45 La estructura de sintaxis ref_piclist_struct(listldx, rplldx, ltrpFlag) está presente en un SPS o en un encabezado de porción. Dependiendo de si la estructura de sintaxis está incluida en un encabezado de porción o en un SPS, se aplica lo siguiente: si está presente en un encabezado de porción, la estructura de sintaxis ref_pic_list_struct(listldx, rplldx, ltrpFlag) especifica la lista listldx de imágenes de referencia de la imagen actual (la imagen que contiene la porción). De lo contrario (presente en un SPS), la estructura de sintaxis ref_pic_list_struct(listldx, rplldx, ltrpFlag) especifica un candidato para la lista listldx de imágenes de referencia, y la expresión "la imagen actual" en la semántica especificada en el resto de esta sección se refiere a cada imagen que 1) tenga una o más porciones que contengan ref_pic_list_idx[listldx] igual a un índice en la lista de las estructuras de sintaxis ref_piclist_struct(listldx, rplldx, ltrpFlag) incluidas en el SPS, y 2) esté en una CVS que tenga el SPS como SPS activo. num_strp_entries[listldx][rplldx] especifica el número de entradas STRP en la estructura de sintaxis ref_pic_list_struct(listldx, rplldx, ltrpFlag). num_ltrp_entries[listldx][rplldx] especifica el número de entradas LTRP en la estructura de sintaxis ref_pic_list_struct(listldx, rplldx, ltrpFlag). Cuando no está presente, se deduce que el valor de num_ltrp_entries[listldx][rplldx] es igual a 0. La variable NumEntriesInList[listldx][rplldx] se obtiene de la siguiente manera: NumEntriesInList[listldx][rplldx] = num_strp_entries[listldx][rplldx] + num_ltrp_entries[listldx][rplldx]. El valor de NumEntriesInList[listldx][rplldx] estará en el intervalo de 0 a sps_max_dec_pic_buffering_minus1, inclusive. It_ref_pic_flag[listldx][i] igual a 1 especifica que la i-ésima entrada en la estructura de sintaxis ref_pic_list_struct(listldx, rplldx, ltrpFlag) es una entrada LTRP.
- 50
- 55
- 60

lt_ref_pic_flag[listIdx][rplIdx][i] igual a 0 especifica que la i-ésima entrada en la estructura de sintaxis ref_pic_list_struct(listIdx, rplIdx, ltrpFlag) es una entrada STRP. Cuando no está presente, se deduce que el valor de lt_ref_pic_flag[listIdx][rplIdx][i] es igual a 0. Es un requisito de conformidad de flujo de bits que la suma de lt_ref_pic_flag[listIdx][rplIdx][i] para todos los valores de i en el intervalo de 0 a NumEntriesInList[listIdx][rplIdx] - 1, inclusive, sea igual a num_ltrp_entries[listIdx][rplIdx]. delta_poc_st[listIdx][rplIdx][i], cuando la i-ésima entrada es la primera entrada STRP en la estructura de sintaxis ref_pic_list_struct(rplIdx, ltrpFlag), especifica la diferencia entre los valores de recuento de orden de imágenes de la imagen actual y la imagen a la que hace referencia la i-ésima entrada o, cuando la i-ésima entrada es una entrada STRP, pero no la primera entrada STRP, en la estructura de sintaxis ref_pic_list_struct(rplIdx, ltrpFlag), especifica la diferencia entre los valores de recuento de orden de imágenes de las imágenes a las que hacen referencia la i-ésima entrada y la entrada STRP anterior en la estructura de sintaxis ref_pic_list_struct(listIdx, rplIdx, ltrpFlag). El valor de delta_poc_st[listIdx][rplIdx][i] estará en el intervalo de -215 a 215 - 1, inclusive. poc_lsb_lt[listIdx][rplIdx][i] especifica el valor del módulo de recuento de orden de imágenes MaxLtPicOrderCntLsb de la imagen a la que hace referencia la i-ésima entrada en la estructura de sintaxis ref_pic_list_struct(listIdx, rplIdx, ltrpFlag). La longitud del elemento de sintaxis poc_lsb_lt[listIdx][rplIdx][i] es $\text{Log}_2(\text{MaxLtPicOrderCntLsb})$ bits.

Se expone el proceso de decodificación. El proceso de decodificación funciona de la siguiente manera para la imagen actual CurrPic. La decodificación de las unidades NAL se especifica posteriormente. Los procesos expuestos posteriormente especifican los siguientes procesos de decodificación utilizando elementos de sintaxis en la capa de encabezado de porción y superiores. Se obtienen variables y funciones relacionadas con el recuento de orden de imágenes. Es necesario invocar esto sólo para la primera porción de una imagen. Al comienzo del proceso de decodificación para cada porción de una imagen no-IRAP, se invoca el proceso de decodificación para la construcción de listas de imágenes de referencia para la obtención de la lista 0 de imágenes de referencia (RefPicList[0]) y la lista 1 de imágenes de referencia (RefPicList[1]). Se invoca el proceso de decodificación para el marcado de imágenes de referencia, en donde las imágenes de referencia se marcan como "no utilizadas para referencia" o "utilizadas para referencia a largo plazo". Es necesario invocar esto sólo para la primera porción de una imagen. Se invocan los procesos de decodificación para codificar unidades de árbol, cambiar la escala, transformar, filtrar en bucle, etc. Después de que se hayan decodificado todas las porciones de la imagen actual, la imagen decodificada actual se marca como "utilizada para referencia a corto plazo".

Se expone el proceso de decodificación de la unidad NAL. Las entradas a este proceso son unidades NAL de la imagen actual y sus unidades NAL no-VCL asociadas. Las salidas de este proceso son las estructuras de sintaxis de RBSP analizadas encapsuladas dentro de las unidades NAL. El proceso de decodificación para cada unidad NAL extrae la estructura de sintaxis de RBSP de la unidad NAL y, a continuación, analiza la estructura de sintaxis de RBSP.

Se expone el proceso de decodificación de porciones, incluido el proceso de decodificación para el recuento de orden de imágenes. La salida de este proceso es PicOrderCntVal, el recuento de orden de imágenes de la imagen actual. Los recuentos de orden de imágenes se utilizan para identificar imágenes, para obtener parámetros de movimiento en modo de fusión y la predicción de vectores de movimiento, y para la verificación de conformidad del decodificador. Cada imagen codificada está asociada con una variable de recuento de orden de imágenes, denominada PicOrderCntVal. Cuando la imagen actual no es una imagen IRAP, las variables prevPicOrderCntLsb y prevPicOrderCntMsb se obtienen de la siguiente manera: sea prevTid0Pic la imagen anterior en orden de decodificación que tenga TemporalId igual a 0. La variable prevPicOrderCntLsb se establece como igual a slice_pic_order_cnt_lsb de prevTid0Pic. La variable prevPicOrderCntMsb se establece como igual a PicOrderCntMsb de prevTid0Pic.

La variable PicOrderCntMsb de la imagen actual se obtiene de la siguiente manera: si la imagen actual es una imagen IRAP, PicOrderCntMsb se establece como igual a 0. De lo contrario, PicOrderCntMsb se obtiene de la siguiente manera:

```

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

```

PicOrderCntVal se obtiene de la siguiente manera: $\text{PicOrderCntVal} = \text{PicOrderCntMsb} + \text{slice_pic_order_cnt_lsb}$.

Todas las imágenes IRAP tendrán PicOrderCntVal igual a 0, ya que slice_pic_order_cnt_lsb se deduce que es 0 para las imágenes IRAP y prevPicOrderCntLsb y prevPicOrderCntMsb se establecen ambos como iguales a 0. El valor de PicOrderCntVal estará en el intervalo de -231 a 231 - 1, inclusive. En una CVS, los valores de PicOrderCntVal para dos imágenes codificadas cualesquiera no serán los mismos.

En cualquier momento durante el proceso de decodificación, los valores de PicOrderCntVal & (MaxLtPicOrderCntLsb

- 1) para dos imágenes de referencia cualesquiera en el DPB no serán los mismos. La función PicOrderCnt(picX) se especifica de la siguiente manera: PicOrderCnt(picX) = PicOrderCntVal de la imagen picX. La función DiffPicOrderCnt(picA, picB) se especifica de la siguiente manera: DiffPicOrderCnt(picA, picB) = PicOrderCnt(picA) - PicOrderCnt(picB). El flujo de bits no contendrá datos que den como resultado valores de DiffPicOrderCnt(picA, picB) utilizados en el proceso de decodificación que no estén en el intervalo de -215 a 215 - 1, inclusive. Sea X la imagen actual y sean Y y Z otras dos imágenes en la misma secuencia de vídeo codificada (CVS, por sus siglas en inglés), Y y Z se considera que están en la misma dirección de orden de salida de X cuando DiffPicOrderCnt(X, Y) y DiffPicOrderCnt(X, Z) son ambos positivos o ambos negativos.

Se expone el proceso de decodificación para la construcción de listas de imágenes de referencia. Este proceso se invoca al principio del proceso de decodificación para cada porción de una imagen no-IRAP. Las imágenes de referencia se direccionan a través de índices de referencia. Un índice de referencia es un índice en una lista de imágenes de referencia. Al decodificar una porción I, no se utiliza ninguna lista de imágenes de referencia en la decodificación de los datos de la porción. Al decodificar una porción P, sólo se utiliza la lista 0 de imágenes de referencia (es decir, RefPicList[0]) en la decodificación de los datos de la porción. Al decodificar una porción B, tanto la lista 0 de imágenes de referencia como la lista 1 de imágenes de referencia (es decir, RefPicList[1]) se utilizan en la decodificación de los datos de la porción. Al comienzo del proceso de decodificación para cada porción de una imagen no-IRAP, se obtienen las listas RefPicList[0] y RefPicList[1] de imágenes de referencia. Las listas de imágenes de referencia se utilizan en el marcado de imágenes de referencia o en la decodificación de los datos de la porción. Para una porción I de una imagen no-IRAP que no sea la primera porción de la imagen, RefPicList[0] y RefPicList[1] se obtienen con el propósito de verificar la conformidad del flujo de bits, pero su obtención no es necesaria para decodificar la o las imágenes actuales que siguen a la imagen actual en orden de decodificación. Para una porción P que no sea la primera porción de una imagen, RefPicList[1] se obtiene con el propósito de verificar la conformidad del flujo de bits, pero su obtención no es necesaria para decodificar la o las imágenes actuales que siguen a la imagen actual en orden de decodificación. Las listas RefPicList[0] y RefPicList[1] de imágenes de referencia se construyen de la siguiente manera:

```
for( i = 0; i < 2; i++ ) {
    if( ref_pic_list_sps_flag[i] ) {
        RplIdx[i] = ref_pic_list_idx[i]
    } else {
        RplIdx[i] = num_ref_pic_lists_in_sps[i]
    }
    for( j = 0; picBase = PicOrderCntVal( j < NumEntriesInList( i ][ RplIdx[i] ][ j ] ) ) {
        if( it_ref_pic_flag[i][ RplIdx[i] ][ j ] ) {
            RefPicList[i][ j ] = picBase + delta_pic_ord[i][ RplIdx[i] ][ j ]
            if( hay una imagen pic de referencia en el DPB con PicOrderCntVal igual a RefPicList[i][ j ] ) {
                RefPicList[i][ j ] = picA
            } else {
                RefPicList[i][ j ] = "ninguna imagen de referencia"
            }
            picBase = RefPicList[i][ j ]
        } else {
            if( hay una pic de referencia en el DPB con PicOrderCntVal & ( MaxPicOrderCntList[i] - 1 )
                igual a picBase + delta_pic_ord[i][ RplIdx[i] ][ j ] ) {
                RefPicList[i][ j ] = picA
            } else {
                RefPicList[i][ j ] = "ninguna imagen de referencia"
            }
        }
    }
}
```

Para cada i igual a 0 o 1, se aplica lo siguiente: las primeras entradas NumRefIdxActive[i] en RefPicList[i] se denominan entradas activas en RefPicList[i], y las otras entradas en RefPicList[i] se denominan entradas inactivas en RefPicList[i]. Cada entrada en RefPicList[i][j] para j en el intervalo de 0 a NumEntriesInList[i][RplIdx[i]] - 1, inclusive, se conoce como una entrada STRP si it_ref_pic_flag[i][RplIdx[i]][j] es igual a 0, y como una entrada LTRP de lo contrario. Es posible que a una imagen en particular hagan referencia tanto una entrada en RefPicList[0] como una entrada en RefPicList[1]. También es posible que a una imagen en particular hagan referencia más de una entrada en RefPicList[0] o más de una entrada en RefPicList[1]. Las entradas activas en RefPicList[0] y las entradas activas en RefPicList[1] se refieren colectivamente a todas las imágenes de referencia que se utilizan para la interpredicción de la imagen actual y una o más imágenes que siguen a la imagen actual en orden de decodificación. Las entradas inactivas en RefPicList[0] y las entradas inactivas en RefPicList[1] se refieren colectivamente a todas las imágenes de referencia que no se utilizan para la interpredicción de la imagen actual, pero se utilizan para la interpredicción de una o más imágenes que siguen a la imagen actual en orden de decodificación. Hay una o más entradas en RefPicList[0] o RefPicList[1] que son iguales a "ninguna imagen de referencia" porque las imágenes correspondientes no están presentes en el DPB. Cada entrada inactiva en RefPicList[0] o RefPicList[0] que sea igual a "ninguna imagen de referencia" debe ignorarse. Se debe deducir una pérdida de imagen no intencionada para cada entrada activa en RefPicList[0] o RefPicList[1] que sea igual a "ninguna imagen de referencia".

Es un requisito de conformidad de flujo de bits que se apliquen las siguientes restricciones: para cada i igual a 0 o 1,

NumEntriesInList[i][RplIdx[i]] no será menor que NumRefIdxActive[i]. La imagen a la que haga referencia cada entrada activa en RefPicList[0] o RefPicList[1] estará presente en el DPB y tendrá TemporalId menor o igual que el de la imagen actual. Opcionalmente, se especifica además la siguiente restricción: el índice de entrada de cualquier entrada inactiva en RefPicList[0] o RefPicList[1] no se utilizará como índice de referencia para la decodificación de la imagen actual. Opcionalmente, se especifica además la siguiente restricción: una entrada inactiva en RefPicList[0] o RefPicList[1] no se referirá a la misma imagen que cualquier otra entrada en RefPicList[0] o RefPicList[1]. Una entrada STRP en RefPicList[0] o RefPicList[1] de una porción de una imagen y una entrada LTRP en RefPicList[0] o RefPicList[1] de la misma porción o de una porción diferente de la misma imagen no se referirán a la misma imagen. A la imagen actual en sí misma no hará referencia ninguna entrada en RefPicList[0] o RefPicList[1]. No habrá ninguna entrada LTRP en RefPicList[0] o RefPicList[1] para la que la diferencia entre PicOrderCntVal de la imagen actual y PicOrderCntVal de la imagen a la que hace referencia la entrada sea superior o igual a 224. Sea setOfRefPics el conjunto de imágenes únicas a las que hacen referencia todas las entradas de RefPicList[0] y todas las entradas de RefPicList[1]. El número de imágenes en setOfRefPics será menor o igual que sps_max_dec_pic_buffering_minus1, y setOfRefPics será el mismo para todas las porciones de una imagen.

Proceso de decodificación para el marcado de imágenes de referencia.

Este proceso se invoca una vez por imagen, después de decodificar un encabezado de porción y el proceso de decodificación para la construcción de listas de imágenes de referencia para la porción, pero antes de la decodificación de los datos de la porción. Este proceso da como resultado que una o más imágenes de referencia en el DPB se marquen como "no utilizadas para referencia" o "utilizadas para referencia a largo plazo". Una imagen decodificada en el DPB puede marcarse como "no utilizada para referencia", "utilizada para referencia a corto plazo" o "utilizada para referencia a largo plazo", pero sólo una de estas tres en cualquier momento dado durante la operación del proceso de decodificación. Asignar una de estas marcas a una imagen elimina implícitamente otra de estas marcas cuando corresponda. Cuando se hace referencia a una imagen como marcada como "utilizada para referencia", esto se refiere colectivamente a que la imagen está marcada como "utilizada para referencia a corto plazo" o "utilizada para referencia a largo plazo" (pero no ambas). Cuando la imagen actual es una imagen IRAP, todas las imágenes de referencia actualmente presentes en el DPB (si las hay) se marcan como "no utilizadas para referencia". Las STRP se identifican por sus valores PicOrderCntVal. Las LTRP se identifican mediante los $\text{Log}_2(\text{MaxLtPicOrderCntLsb})$ LSB de sus valores PicOrderCntVal. Se aplica lo siguiente: para cada entrada LTRP en RefPicList[0] o RefPicList[1], cuando la imagen a la que se hace referencia es una STRP, la imagen se marca como "utilizada para referencia a largo plazo". Cada imagen de referencia en el DPB a la que no haga referencia ninguna entrada de RefPicList[0] o RefPicList[1] está marcada como "no utilizada para referencia".

Se proporciona una descripción detallada de la segunda realización de la divulgación. Esta sección documenta una segunda realización de la divulgación como se describió anteriormente. La descripción es relativa al último WD de VVC. En esta realización, se señala en el SPS un conjunto de estructuras de lista de imágenes de referencia, compartido por la lista 0 de imágenes de referencia y la lista 1 de imágenes de referencia.

Sintaxis de RBSP de conjunto de parámetros de secuencia.

seq_parameter_set_rbsp() {	Descriptor
sps_seq_parameter_set_id	ue(v)
chroma_format_idc	ue(v)
if(chroma_format_idc == 3)	
separate_colour_plane_flag	u(1)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
bit_depth_luma_minus8	ue(v)
bit_depth_chroma_minus8	ue(v)
log2_max_pic_order_cnt_lsb_minus4	ue(v)
sps_max_dec_pic_buffering_minus1	ue(v)
qtbtt_dual_tree_intra_flag	ue(v)
log2_ctu_size_minus2	ue(v)
log2_min_qt_size_intra_slices_minus2	ue(v)
log2_min_qt_size_inter_slices_minus2	ue(v)
max_mtt_hierarchy_depth_inter_slices	ue(v)
max_mtt_hierarchy_depth_intra_slices	ue(v)
num_ref_pic_lists_in_sps	ue(v)
long_term_ref_pics_flag	u(1)
if(long_term_ref_pics_flag)	
additional_lt_poc_lsb	ue(v)
for(i = 0; i < num_ref_pic_lists_in_sps; i++)	
ref_picliststruct(i, long_term_ref_pics_flag)	
rbbsp_trailing_bits()	

```
} 
```

Sintaxis de RBSP de conjunto de parámetros de imagen.

pic_parameter_set_rbsp() {	Descriptor
pps_pic_parameter_set_id	ue(v)
pps_seq_parameter_set_id	ue(v)
for(i = 0; i < 2; i++)	
num_ref_idx_default_active_minus1[i]	ue(v)
rbps_trailing_bits()	
}	

5 Sintaxis de encabezado de porción.

slice_header() {	Descriptor
slice_pic_parameter_set_id	ue(v)
slice_address	u(v)
slice_type	ue(v)
if (slice_type != I)	
log2_diff_ctu_max_bt_size	ue(v)
if(nal_unit_type != IRAP_NUT) {	
slice_pic_order_cnt_lsb	u(v)
for(i = 0; i < 2; i++) {	
ref_pic_list_sps_flag[i]	u(1)
if(ref_pic_list_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps > 1)	
ref_pic_list_idx[i]	u(v)
} else	
ref_piclist_struct(num_ref_pic_lists_in_sps + i, long_term_ref_pics_flag)	
}	
if(slice_type == P slice_type == B) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag)	
for(i = 0; i < (slice_type == B ? 2 : 1); i++)	
num_ref_idx_active_minus1[i]	ue(v)
}	
}	
byte_alignment()	
}	

Sintaxis de estructura de lista de imágenes de referencia.

ref_pic_list_struct(rplsIdx, ltrpFlag) {	Descriptor
num_strp_entries[rplsIdx]	ue(v)
if(ltrpFlag)	
num_ltrp_entries[rplsIdx]	ue(v)
for(i = 0; i < NumEntriesInList[rplsIdx]; i++) {	
if(num_ltrp_entries[rplsIdx] > 0)	
lt_ref_pic_flag[rplsIdx][i]	
if(!lt_ref_pic_flag[rplsIdx][i])	
delta_poc_st[rplsIdx][i] // No ha de tener _minus1 aquí, para permitir que la misma ref pic esté incluida en múltiples posiciones de una ref pic list. Como efecto secundario, esto también posibilitaría hacer referencia a la imagen actual (es decir, copia intrabloque). Cabe señalar también que está firmado.	se(v)
else	
poc_lsb_lt[rplsIdx][i]	u(v)
}	
}	

10

Se expone la semántica del encabezado de unidad NAL.

Semántica de RBSP de conjunto de parámetros de secuencia.

Un `log2_max_pic_order_cnt_lsb_minus4` especifica el valor de la variable `MaxPicOrderCntLsb` que se utiliza en el proceso de decodificación para el recuento de orden de imágenes de la siguiente manera: $\text{MaxPicOrderCntLsb} = 2(\text{log2_max_pic_order_cnt_lsb_minus4} + 4)$. El valor de `log2_max_pic_order_cnt_lsb_minus4` estará en el intervalo de 0 a 12, inclusive. `sps_max_dec_pic_buffering_minus1` plus 1 especifica el tamaño máximo requerido del búfer de imágenes decodificadas para la CVS en unidades de búferes de almacenamiento de imágenes. El valor de `sps_max_dec_pic_buffering_minus1` estará en el intervalo de 0 a `MaxDpbSize` - 1, inclusive, donde `MaxDpbSize` es como se especifica en otro lugar. `num_ref_pic_lists_in_sps` especifica el número de estructuras de sintaxis `ref_pic_list_struct( rplIdx, ltrpFlag )` incluidas en el SPS. El valor de `num_ref_pic_lists_in_sps` estará en el intervalo de 0 a 128, inclusive. Un decodificador debe asignar memoria para un número total de `num_short_term_ref_pic_sets` + 2 estructuras de sintaxis `ref_pic_list_struct( rplIdx, ltrpFlag )`, ya que hay dos estructuras de sintaxis `ref_pic_list_struct( rplIdx, ltrpFlag )` directamente señalizadas en los encabezados de porción de una imagen actual. `long_term_ref_pics_flag` igual a 0 especifica que no se usa LTRP para la interpredicción de ninguna imagen codificada en la CVS. `long_term_ref_pics_flag` igual a 1 especifica que se usan LTRP para la interpredicción de una o más imágenes codificadas en la CVS. `additional_lt_poc_lsb` especifica el valor de la variable `MaxLtPicOrderCntLsb` que se utiliza en el proceso de decodificación para las listas de imágenes de referencia de la siguiente manera: $\text{MaxLtPicOrderCntLsb} = 2(\text{log2_max_pic_order_cnt_lsb_minus4} + 4 + \text{additional_lt_poc_lsb})$. El valor de `additional_lt_poc_lsb` estará en el intervalo de 0 a $32 - \text{log2_max_pic_order_cnt_lsb_minus4} - 4$, inclusive. Cuando no está presente, se deduce que el valor de `additional_lt_poc_lsb` es igual a 0.

Se expone la semántica de RBSP del conjunto de parámetros de imagen.

Semántica de encabezado de porción.

Cuando estén presentes, el valor de cada uno de los elementos de sintaxis de encabezado de porción `slice_pic_parameter_set_id` y `slice_pic_order_cnt_lsb` será el mismo en todos los encabezados de porción de una imagen codificada. `slice_type` especifica el tipo de codificación de la porción de acuerdo con la Tabla 7-3.

Tabla 7-3 - Asociación de nombres a `slice_type`

<code>slice_type</code>	Nombre de <code>slice_type</code>
0	B (porción B)
1	P (porción P)
2	I (porción I)

Cuando `nal_unit_type` es igual a `IRAP_NUT`, es decir, la imagen es una imagen IRAP, `slice_type` será igual a 2. ... `slice_pic_order_cnt_lsb` especifica el módulo de recuento de orden de imágenes `MaxPicOrderCntLsb` para la imagen actual. La longitud del elemento de sintaxis `slice_pic_order_cnt_lsb` es $\text{log2_max_pic_order_cnt_lsb_minus4} + 4$ bits. El valor de `slice_pic_order_cnt_lsb` estará en el intervalo de 0 a `MaxPicOrderCntLsb` - 1, inclusive. Cuando `slice_pic_order_cnt_lsb` no está presente, se deduce que `slice_pic_order_cnt_lsb` es igual a 0. `ref_pic_list_sps_flag[i]` igual a 1 especifica que la lista *i* de imágenes de referencia de la imagen actual se obtiene sobre la base de una de las estructuras de sintaxis `ref_pic_list_struct( rplIdx, ltrpFlag )` en el SPS activo. `ref_pic_list_sps_flag[i]` igual a 0 especifica que la lista *i* de imágenes de referencia de la imagen actual se obtiene sobre la base de la estructura de sintaxis `ref_pic_list_struct( rplIdx, ltrpFlag )` que se incluye directamente en los encabezados de porción de la imagen actual. Cuando `num_ref_pic_lists_in_sps` es igual a 0, el valor de `ref_pic_list_sps_flag[i]` será igual a 0. `ref_pic_list_idx[i]` especifica el índice, en la lista de las estructuras de sintaxis `ref_pic_list_struct( rplIdx, ltrpFlag )` que se incluyen en el SPS activo, de la estructura de sintaxis `ref_pic_list_struct( rplIdx, ltrpFlag )` que se utiliza para la obtención de la lista *i* de imágenes de referencia de la imagen actual. El elemento de sintaxis `ref_pic_list_idx[i]` está representado por $\text{Ceil}(\text{Log2}(\text{num_ref_pic_lists_in_sps}))$ bits. Cuando no está presente, se deduce que el valor de `ref_pic_list_idx[i]` es igual a 0. El valor de `ref_pic_list_idx[i]` estará en el intervalo de 0 a `num_ref_pic_lists_in_sps` - 1, inclusive. `num_ref_idx_active_override_flag` igual a 1 especifica que el elemento de sintaxis `num_ref_idx_active_minus1[0]` está presente para porciones P y B y que el elemento de sintaxis `num_ref_idx_active_minus1[1]` está presente para porciones B. `num_ref_idx_active_override_flag` igual a 0 especifica que los elementos de sintaxis `num_ref_idx_active_minus1[0]` y `num_ref_idx_active_minus1[1]` no están presentes.

`num_ref_idx_active_minus1[i]`, cuando está presente, especifica el valor de la variable `NumRefIdxActive[i]` de la siguiente manera: $\text{NumRefIdxActive}[i] = \text{num_ref_idx_active_minus1}[i] + 1$. El valor de `num_ref_idx_active_minus1[i]` estará en el intervalo de 0 a 14, inclusive. El valor de `NumRefIdxActive[i]` - 1 especifica el índice de referencia máximo para la lista *i* de imágenes de referencia que se utiliza para decodificar la porción. Cuando el valor de `NumRefIdxActive[i]` es igual a 0, no se utiliza ningún índice de referencia para la lista *i* de imágenes de referencia para decodificar la porción. Para *i* igual a 0 o 1, cuando la porción actual es una porción B y `num_ref_idx_active_override_flag` es igual a 0, se deduce que `NumRefIdxActive[i]` es igual a `num_ref_idx_default_active_minus1[i] + 1`. Cuando la porción actual es una porción P y `num_ref_idx_active_override_flag` es igual a 0, se deduce que `NumRefIdxActive[0]` es igual a `num_ref_idx_default_active_minus1[0] + 1`. Cuando la porción actual es una porción P, se deduce que `NumRefIdxActive[1]` es igual a 0. Cuando la porción actual es una porción I, se deduce que tanto `NumRefIdxActive[0]` como `NumRefIdxActive[1]` son iguales a 0.

como NumRefIdxActive[1] son iguales a 0.

Como alternativa, para i igual a 0 o 1, se aplica lo siguiente después de lo anterior: sea $rplIdx1$ establecido igual a $ref_pic_list_sps_flag[i] ? ref_pic_list_idx[i] : num_ref_pic_lists_in_sps[i]$, y $numRpEntries[i]$ igual a $num_strp_entries[i][rplIdx1] + num_ltrp_entries[i][rplIdx1]$. Cuando $NumRefIdxActive[i]$ es mayor que $numRpEntries[i]$, el valor de $NumRefIdxActive[i]$ se establece igual a $numRpEntries[i]$.

Semántica de estructura de lista de imágenes de referencia.

La estructura de sintaxis $ref_pic_list_struct(rplIdx, ltrpFlag)$ está presente en un SPS o en un encabezado de porción. Dependiendo de si la estructura de sintaxis está incluida en un encabezado de porción o en un SPS, se aplica lo siguiente: si está presente en un encabezado de porción, la estructura de sintaxis $ref_pic_list_struct(rplIdx, ltrpFlag)$ especifica una lista de imágenes de referencia de la imagen actual (la imagen que contiene la porción). De lo contrario (presente en un SPS), la estructura de sintaxis $ref_pic_list_struct(rplIdx, ltrpFlag)$ especifica una lista de imágenes de referencia candidata, y la expresión "la imagen actual" en la semántica especificada en el resto de esta sección se refiere a cada imagen que 1) tenga una o más porciones que contengan $ref_pic_list_idx[i]$ igual a un índice en la lista de las estructuras de sintaxis $ref_pic_list_struct(rplIdx, ltrpFlag)$ incluidas en el SPS, y 2) esté en una CVS que tenga el SPS como SPS activo. $num_strp_entries[rplIdx]$ especifica el número de entradas STRP en la estructura de sintaxis $ref_pic_list_struct(rplIdx, ltrpFlag)$. $num_ltrp_entries[rplIdx]$ especifica el número de entradas LTRP en la estructura de sintaxis $ref_pic_list_struct(rplIdx, ltrpFlag)$. Cuando no está presente, se deduce que el valor de $num_ltrp_entries[rplIdx]$ es igual a 0.

La variable $NumEntriesInList[rplIdx]$ se obtiene de la siguiente manera: $NumEntriesInList[rplIdx] = num_strp_entries[rplIdx] + num_ltrp_entries[rplIdx]$. El valor de $NumEntriesInList[rplIdx]$ estará en el intervalo de 0 a $sps_max_dec_pic_buffering_minus1$, inclusive. $lt_ref_pic_flag[rplIdx][i]$ igual a 1 especifica que la i -ésima entrada en la estructura de sintaxis $ref_pic_list_struct(rplIdx, ltrpFlag)$ es una entrada LTRP. $lt_ref_pic_flag[rplIdx][i]$ igual a 0 especifica que la i -ésima entrada en la estructura de sintaxis $ref_pic_list_struct(rplIdx, ltrpFlag)$ es una entrada STRP. Cuando no está presente, se deduce que el valor de $lt_ref_pic_flag[rplIdx][i]$ es igual a 0. Es un requisito de conformidad de flujo de bits que la suma de $lt_ref_pic_flag[rplIdx][i]$ para todos los valores de i en el intervalo de 0 a $NumEntriesInList[rplIdx] - 1$, inclusive, sea igual a $num_ltrp_entries[rplIdx]$. $delta_poc_st[rplIdx][i]$, cuando la i -ésima entrada es la primera entrada STRP en la estructura de sintaxis $ref_pic_list_struct(rplIdx, ltrpFlag)$, especifica la diferencia entre los valores de recuento de orden de imágenes de la imagen actual y la imagen a la que hace referencia la i -ésima entrada o, cuando la i -ésima entrada es una entrada STRP, pero no la primera entrada STRP, en la estructura de sintaxis $ref_pic_list_struct(rplIdx, ltrpFlag)$, especifica la diferencia entre los valores de recuento de orden de imágenes de las imágenes a las que hacen referencia la i -ésima entrada y la entrada STRP anterior en la estructura de sintaxis $ref_pic_list_struct(rplIdx, ltrpFlag)$. El valor de $delta_poc_st[rplIdx][i]$ estará en el intervalo de 0 a 215 - 1, inclusive. $poc_lsb_lt[rplIdx][i]$ especifica el valor del módulo de recuento de orden de imágenes $MaxLtPicOrderCntLsb$ de la imagen a la que hace referencia la i -ésima entrada en la estructura de sintaxis $ref_pic_list_struct(rplIdx, ltrpFlag)$. La longitud del elemento de sintaxis $poc_lsb_lt[rplIdx][i]$ es $\log_2(MaxLtPicOrderCntLsb)$ bits.

Se aplica el proceso de decodificación general especificado como parte de la descripción detallada de la primera realización de esta divulgación. Se describe el proceso de decodificación de la unidad NAL. Se aplica el proceso de decodificación de la unidad NAL especificado como parte de la descripción detallada de la primera realización de esta divulgación.

Se proporciona el proceso de decodificación de porciones.

Proceso de decodificación para el recuento de orden de imágenes.

Se aplica el proceso de decodificación para el recuento de orden de imágenes especificado como parte de la descripción detallada de la primera realización de esta divulgación.

Proceso de decodificación para la construcción de listas de imágenes de referencia.

Este proceso se invoca al principio del proceso de decodificación para cada porción de una imagen no-IRAP. Las imágenes de referencia se direccionan a través de índices de referencia. Un índice de referencia es un índice en una lista de imágenes de referencia. Al decodificar una porción I, no se utiliza ninguna lista de imágenes de referencia en la decodificación de los datos de la porción. Al decodificar una porción P, sólo se utiliza la lista 0 de imágenes de referencia (es decir, $RefPicList[0]$) en la decodificación de los datos de la porción. Al decodificar una porción B, tanto la lista 0 de imágenes de referencia como la lista 1 de imágenes de referencia (es decir, $RefPicList[1]$) se utilizan en la decodificación de los datos de la porción. Al comienzo del proceso de decodificación para cada porción de una imagen no-IRAP, se obtienen las listas $RefPicList[0]$ y $RefPicList[1]$ de imágenes de referencia. Las listas de imágenes de referencia se utilizan en el marcado de imágenes de referencia o en la decodificación de los datos de la porción. Para una porción I de una imagen no-IRAP que no sea la primera porción de la imagen, $RefPicList[0]$ y $RefPicList[1]$ se obtienen con el propósito de verificar la conformidad del flujo de bits, pero su obtención no es

necesaria para decodificar la o las imágenes actuales que siguen a la imagen actual en orden de decodificación. Para una porción P que no sea la primera porción de una imagen, RefPicList[1] se obtiene con el propósito de verificar la conformidad del flujo de bits, pero su obtención no es necesaria para decodificar la o las imágenes actuales que siguen a la imagen actual en orden de decodificación.

5

Las listas RefPicList[0] y RefPicList[1] de imágenes de referencia se construyen de la siguiente manera:

```
for( i = 0; i < 2; i++) {
    if( ref_pic_list_sps_flag[i] )
        RplIdx[i] = ref_pic_list_idx[i]
    else
        RplIdx[i] = num_ref_pic_lists_in_sps
    for( j = 0; pic_base = PicOrderCntVal, j < NumEntriesInList[ RplIdx[i] ]; j++) {
        if( ! ( ref_pic_flag[ RplIdx[i] ][ j ] ) ) {
            RefPicList[i][ j ] = pic_base + delta_poc_8[ RplIdx[i] ][ j ]
            if( hay una imagen picA de referencia en el DPB con PicOrderCntVal igual a RefPicList[i][ j ] )
                RefPicList[i][ j ] = picA
            else
                RefPicList[i][ j ] = "ninguna imagen de referencia"
            pic_base = RefPicList[i][ j ]
        } else {
            if( hay una picA de referencia en el DPB con PicOrderCntVal & ( MaxLTPicOrderCntList - 1 )
                igual a poc_sub_8[ RplIdx[i] ][ j ] )
                RefPicList[i][ j ] = picA
            else
                RefPicList[i][ j ] = "ninguna imagen de referencia"
        }
    }
}
```

- 10 Para cada i igual a 0 o 1, se aplica lo siguiente: las primeras entradas NumRefIdxActive[i] en RefPicList[i] se denominan entradas activas en RefPicList[i], y las otras entradas en RefPicList[i] se denominan entradas inactivas en RefPicList[i]. Cada entrada RefPicList[i][j] para j en el intervalo de 0 a NumEntriesInList[RplIdx[i]] - 1, inclusive, se conoce como una entrada STRP si lt_ref_pic_flag[RplIdx[i]][j] es igual a 0, y como una entrada LTRP de lo contrario. Es posible que a una imagen en particular hagan referencia tanto una entrada en RefPicList[0] como una
- 15 entrada en RefPicList[1]. También es posible que a una imagen en particular hagan referencia más de una entrada en RefPicList[0] o más de una entrada en RefPicList[1]. Las entradas activas en RefPicList[0] y las entradas activas en RefPicList[1] se refieren colectivamente a todas las imágenes de referencia que se utilizan para la interpredicción de la imagen actual y una o más imágenes que siguen a la imagen actual en orden de decodificación. Las entradas inactivas en RefPicList[0] y las entradas inactivas en RefPicList[1] se refieren colectivamente a todas las imágenes
- 20 de referencia que no se utilizan para la interpredicción de la imagen actual, pero se utilizan para la interpredicción de una o más imágenes que siguen a la imagen actual en orden de decodificación. Hay una o más entradas en RefPicList[0] o RefPicList[1] que son iguales a "ninguna imagen de referencia" porque las imágenes correspondientes no están presentes en el DPB. Cada entrada inactiva en RefPicList[0] o RefPicList[1] que sea igual a "ninguna imagen de referencia" debe ignorarse. Se debe deducir una pérdida de imagen no intencionada para cada entrada
- 25 activa en RefPicList[0] o RefPicList[1] que sea igual a "ninguna imagen de referencia".

Es un requisito de conformidad de flujo de bits que se apliquen las siguientes restricciones: para cada i igual a 0 o 1, NumEntriesInList[RplIdx[i]] no será menor que NumRefIdxActive[i]. La imagen a la que haga referencia cada entrada activa en RefPicList[0] o RefPicList[1] estará presente en el DPB y tendrá TemporalId menor o igual que el

30 de la imagen actual. Opcionalmente, se especifica además la siguiente restricción: el índice de entrada de cualquier entrada inactiva en RefPicList[0] o RefPicList[1] no se utilizará como índice de referencia para la decodificación de la imagen actual. Opcionalmente, se especifica además la siguiente restricción: una entrada inactiva en RefPicList[0] o RefPicList[1] no se referirá a la misma imagen que cualquier otra entrada en RefPicList[0] o RefPicList[1]. Una entrada STRP en RefPicList[0] o RefPicList[1] de una porción de una imagen y una entrada LTRP en RefPicList[0]

35 o RefPicList[1] de la misma porción o de una porción diferente de la misma imagen no se referirán a la misma imagen. A la imagen actual en sí misma no hará referencia ninguna entrada en RefPicList[0] o RefPicList[1]. No habrá ninguna entrada LTRP en RefPicList[0] o RefPicList[1] para la que la diferencia entre PicOrderCntVal de la imagen actual y PicOrderCntVal de la imagen a la que hace referencia la entrada sea superior o igual a 224. Sea setOfRefPics el conjunto de imágenes únicas a las que hacen referencia todas las entradas de RefPicList[0] y todas las entradas

40 de RefPicList[1]. El número de imágenes en setOfRefPics será menor o igual que sps_max_dec_pic_buffering_minus1, y setOfRefPics será el mismo para todas las porciones de una imagen.

Se expone el proceso de decodificación para el marcado de imágenes de referencia.

- 45 Este proceso se invoca una vez por imagen, después de decodificar un encabezado de porción y el proceso de decodificación para la construcción de listas de imágenes de referencia para la porción, pero antes de la decodificación

de los datos de la porción. Este proceso da como resultado que una o más imágenes de referencia en el DPB se marquen como "no utilizadas para referencia" o "utilizadas para referencia a largo plazo". Una imagen decodificada en el DPB puede marcarse como "no utilizada para referencia", "utilizada para referencia a corto plazo" o "utilizada para referencia a largo plazo", pero sólo una de estas tres en cualquier momento dado durante la operación del proceso de decodificación. Asignar una de estas marcas a una imagen elimina implícitamente otra de estas marcas cuando corresponda. Cuando se hace referencia a una imagen como marcada como "utilizada para referencia", esto se refiere colectivamente a que la imagen está marcada como "utilizada para referencia a corto plazo" o "utilizada para referencia a largo plazo" (pero no ambas). Cuando la imagen actual es una imagen IRAP, todas las imágenes de referencia actualmente presentes en el DPB (si las hay) se marcan como "no utilizadas para referencia". Las STRP se identifican por sus valores PicOrderCntVal. Las LTRP se identifican mediante los $\text{Log2}(\text{MaxLtPicOrderCntLsb})$ LSB de sus valores PicOrderCntVal.

Se aplica lo siguiente: para cada entrada LTRP en RefPicList[0] o RefPicList[1], cuando la imagen a la que se hace referencia es una STRP, la imagen se marca como "utilizada para referencia a largo plazo". Cada imagen de referencia en el DPB a la que no haga referencia ninguna entrada de RefPicList[0] o RefPicList[1] está marcada como "no utilizada para referencia".

La Figura 5 es una realización de un método 500 de decodificación de un flujo de bits de vídeo codificado implementado por un decodificador de vídeo (por ejemplo, el decodificador 30 de vídeo). El método 500 puede realizarse después de que el flujo de bits decodificado haya sido recibido directa o indirectamente desde un codificador de vídeo (por ejemplo, el codificador 20 de vídeo). El método 500 se realiza para mejorar el proceso de decodificación (por ejemplo, hacer el proceso de decodificación más eficaz, más rápido, etc., que los procesos de decodificación convencionales). Por lo tanto, como cuestión práctica, se puede mejorar el rendimiento de un códec, lo que conduce a una mejor experiencia de usuario.

En el bloque 502, se analiza un encabezado de porción de una porción actual representada en el flujo de bits de vídeo codificado. En una realización, el encabezado de porción comprende una estructura de lista de imágenes de referencia. En una realización, un orden de entradas en la estructura de lista de imágenes de referencia es el mismo que un orden de imágenes de referencia correspondientes en la lista de imágenes de referencia. En una realización, el orden es de cero a un valor indicado. En una realización, el valor indicado es de cero a un valor indicado por `sps_max_dec_pic_buffering_minus1`.

En el bloque 504, se obtiene una lista de imágenes de referencia de la porción actual sobre la base de la estructura de lista de imágenes de referencia. En una realización, la lista de imágenes de referencia se denomina RefPicList[0] o RefPicList[1].

En el bloque 506, se obtiene al menos un bloque reconstruido de la porción actual sobre la base de la lista de imágenes de referencia. En una realización, el al menos un bloque reconstruido se utiliza para generar una imagen mostrada en una pantalla de un dispositivo electrónico.

En una realización, la lista de imágenes de referencia comprende una lista de imágenes de referencia utilizadas para la interpredicción. En una realización, la interpredicción es para una porción P o para una porción B.

En una realización, el encabezado de porción contiene un señalizador de conjunto de parámetros de secuencia (SPS) de lista de imágenes de referencia denominado `ref_pic_list_sps_flag[ i ]`. Cuando este señalizador es igual a 1, la *i*-ésima lista de imágenes de referencia, es decir, RefPicList[*i*], no se señala directamente en el encabezado de porción, sino que se hace referencia a la misma desde el SPS. Cuando este señalizador es igual a 0, la *i*-ésima lista de imágenes de referencia, es decir, RefPicList[*i*], se señala directamente en el encabezado de porción y no se hace referencia a la misma desde el SPS. En una realización, el encabezado de porción contiene un señalizador de anulación activa de índice de referencia numérica designado mediante `num_ref_idx_active_override_flag`. Cuando este señalizador es igual a 1, el número de entradas activas en cada lista de imágenes de referencia es el valor por defecto señalado en el PPS. Cuando este señalizador es igual a 0, el número de entradas activas en cada lista de imágenes de referencia es el señalado explícitamente en el encabezado de porción.

En una realización, la lista de imágenes de referencia se denomina RefPicList[0] o RefPicList[1], y un orden de entradas en la estructura de lista de imágenes de referencia es el mismo que un orden de imágenes de referencia correspondientes en la lista de imágenes de referencia.

Se proporciona un resumen de realizaciones alternativas basadas en la primera y la segunda realizaciones.

Esta sección proporciona breves resúmenes de otras realizaciones alternativas de la divulgación. Los resúmenes son relativos a la descripción de la primera realización. Sin embargo, el concepto básico de la divulgación para las siguientes realizaciones alternativas también es aplicable para la implementación sobre la divulgación para la segunda realización. Tal implementación es con el mismo espíritu de cómo los aspectos se implementan sobre la primera realización.

Semántica del delta POC de las entradas de imágenes de referencia a corto plazo.

En una realización alternativa de la divulgación, la semántica del elemento de sintaxis que especifica el delta POC de la i-ésima entrada en una estructura de lista de imágenes de referencia `ref_pic_list_struct( )` se define como la diferencia de POC entre la imagen actual y la imagen de referencia asociada con esa i-ésima entrada. Parte de la descripción utilizada en la presente memoria es relativa al presente borrador estándar (por ejemplo, el borrador de trabajo de VVC) donde sólo se muestra o describe el delta. El texto eliminado se indica mediante tachadura y cualquier texto agregado está resaltado.

- 5
- 10 La semántica de `delta_poc_st[ listIdx ][ rplIdx ][ i ]` se define de la siguiente manera: `delta_poc_st[ listIdx ][ rplIdx ][ i ]` especifica la diferencia entre los valores de recuento de orden de imágenes de la imagen actual y la imagen a la que hace referencia la i-ésima entrada. El valor de `delta_poc_st[ listIdx ][ rplIdx ][ i ]` estará en el intervalo de -215 a 215 - 1, inclusive.
- 15 Es necesario actualizar la ecuación en el proceso de construcción de la lista de imágenes de referencia. Las listas `RefPicList[ 0 ]` y `RefPicList[ 1 ]` de imágenes de referencia se construyen de la siguiente manera:

```

for( i = 0; i < 2; i++ ) {
    if( ref_pic_list_sps_flag[ i ] )
        RplIdx[ i ] = ref_pic_list_idx[ i ]
    else
        RplIdx[ i ] = num_ref_pic_list_in_sps[ i ]
    for( j = 0; j < NumRefPicListInList[ i ][ RplIdx[ i ] ], j++ ) {
        if( M_ref_pic_flag[ i ][ RplIdx[ i ] ][ j ] ) {
            RefPicOrderList[ i ][ j ] = PicOrderCntVal + delta_poc_st[ i ][ RplIdx[ i ] ][ j ]
            if( hay una imagen picA de referencia en el DPB con PicOrderCntVal igual a RefPicOrderList[ i ][ j ] )
                RefPicList[ i ][ j ] = picA
            else
                RefPicList[ i ][ j ] = "ninguna imagen de referencia"
        } else {
            if( hay una picA de referencia en el DPB con PicOrderCntVal <= { MaxLTPicOrderCntLsb - 1 }
                igual a poc_lsb_R[ i ][ RplIdx[ i ] ][ j ] )
                RefPicList[ i ][ j ] = picA
            else
                RefPicList[ i ][ j ] = "ninguna imagen de referencia"
        }
    }
}

```

- 20 Señalización de entradas de imágenes de referencia a largo plazo.

En una realización alternativa de la divulgación, las entradas de imagen de referencia a largo plazo no se señalizan en la misma estructura de lista de imágenes de referencia que contiene entradas de imagen de referencia a corto plazo. Las entradas de imagen de referencia a largo plazo se señalizan en una estructura separada y para cada entrada en la estructura hay un elemento de sintaxis que describe la posición prevista de la entrada de imagen de referencia a largo plazo para la obtención del índice de entrada correspondiente en la lista de imágenes de referencia final.

- 25

Sintaxis de RBSP de conjunto de parámetros de secuencia.

- 30

seq_parameter_set_rbsp() {	Descriptor
sps_seq_parameter_set_id	ue(v)
chroma_format_idc	ue(v)
if(chroma_format_idc == 3)	
separate_colour_plane_flag	u(1)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
bit_depth_luma_minus8	ue(v)
bit_depth_chroma_minus8	ue(v)
log2_max_pic_order_cnt_lsb_minus4	ue(v)
sps_max_dec_pic_buffering_minus1	ue(v)
qtbtt_dual_tree_intra_flag	ue(v)
log2_ctu_size_minus2	ue(v)
log2_min_qt_size_intra_slices_minus2	ue(v)
log2_min_qt_size_inter_slices_minus2	ue(v)

max_mtt_hierarchy_depth_inter_slices	ue(v)
max_mtt_hierarchy_depth_intra_slices	ue(v)
for(i = 0; i < 2; i++) {	
num_ref_pic_lists_in_sps[i]	ue(v)
for(j = 0; j < num_ref_pic_lists_in_sps[i]; j++)	
ref_pic_list_struct(i, j)	
}	
long_term_ref_pics_flag	u(1)
if(long_term_ref_pics_flag) {	
additional_lt_poc_lsb	ue(v)
num_ref_pic_lists_lt_in_sps	ue(v)
}	
for(i = 0; i < num_ref_pic_lists_lt_in_sps; i++)	
ref_pic_list_lt_struct(i)	
rbsp_trailing_bits()	
}	

Sintaxis de encabezado de porción.

slice_header() {	Descriptor
slice_pic_parameter_set_id	ue(v)
slice_address	u(v)
slice_type	ue(v)
if(slice_type != I)	
log2_diff_ctu_max_bt_size	ue(v)
if(nal_unit_type != IRAP_NUT) {	
slice_pic_order_cnt_lsb	u(v)
for(i = 0; i < 2; i++) {	
ref_pic_list_sps_flag[i]	u(1)
if(ref_pic_list_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1)	
ref_pic_list_idx[i]	u(v)
if(long_term_ref_pics_flag)	
ref_pic_list_lt_idx[i]	u(v)
} else {	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i])	
if(long_term_ref_pics_flag)	
ref_pic_list_lt_struct(num_ref_pic_lists_lt_in_sps + 1)	
}	
}	
if(slice_type == P slice_type == B) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag)	
for(i = 0; i < (slice_type == B ? 2 : 1); i++)	
num_ref_idx_active_minus1[i]	ue(v)
}	
}	
byte_alignment()	
}	

5 Sintaxis de estructura de lista de imágenes de referencia.

ref_pic_list_struct(listIdx, rplIdx) {	Descriptor
num_strp_entries[listIdx][rplIdx]	ue(v)
for(i = 0; i < NumEntriesInList[listIdx][rplIdx]; i++) {	
delta_poc_st[listIdx][rplIdx][i]	se(v)
}	
}	

Sintaxis de estructura de lista de imágenes de referencia a largo plazo.

ref_pic_list_lt_struct(ltRplIdx) {	Descriptor
num_ltrp_entries[ltRplIdx]	ue(v)

for(i = 0; i < num_ltrp_entries[ltrp_sidx]; i++) {	
poc_lsb_ltrp[ltrp_sidx][i]	u(v)
ltrp_pos_idx [ltrp_sidx][i] // especifica el índice de esta imagen de referencia a largo plazo en este RPL	u(v)
}	
}	

Semántica de RBSP de conjunto de parámetros de secuencia.

- 5 Un num_ref_pic_lists_ltrp_in_sps especifica el número de estructuras de sintaxis ref_pic_list_ltrp_struct(ltrp_sidx) incluidas en el SPS. El valor de num_ref_pic_lists_ltrp_in_sps estará en el intervalo de 0 a 64, inclusive. Cuando no está presente, se deduce que el valor de num_ref_pic_lists_ltrp_in_sps es igual a 0.

Semántica de encabezado de porción.

- 10 Un ref_pic_list_ltrp_idx[i] especifica el índice en la lista de las estructuras de sintaxis ref_pic_list_ltrp_struct(ltrp_sidx) incluidas en el SPS activo que se utiliza para la obtención de la lista i de imágenes de referencia de la imagen actual. El elemento de sintaxis ref_pic_list_ltrp_idx[i] está representado por $\lceil \log_2(\text{num_ref_pic_lists_ltrp_in_sps}) \rceil$ bits. El valor de ref_pic_list_ltrp_idx estará en el intervalo de 0 a num_ref_pic_lists_ltrp_in_sps - 1, inclusive.

- 15 Semántica de estructura de lista de imágenes de referencia.

- La estructura de sintaxis ref_pic_list_struct(list_idx, rpl_sidx) está presente en un SPS o en un encabezado de porción. Dependiendo de si la estructura de sintaxis está incluida en un encabezado de porción o en un SPS, se aplica lo siguiente: si está presente en un encabezado de porción, la estructura de sintaxis ref_pic_list_struct(list_idx, rpl_sidx) especifica la lista list_idx de imágenes de referencia a corto plazo de la imagen actual (la imagen que contiene la porción). De lo contrario (presente en un SPS), la estructura de sintaxis ref_pic_list_struct(list_idx, rpl_sidx) especifica un candidato para la lista list_idx de imágenes de referencia a corto plazo, y la expresión "la imagen actual" en la semántica especificada en el resto de esta sección se refiere a cada imagen que 1) tenga una o más porciones que contengan ref_pic_list_idx[list_idx] igual a un índice en la lista de las estructuras de sintaxis ref_pic_list_struct(list_idx, rpl_sidx) incluidas en el SPS, y 2) esté en una CVS que tenga el SPS como SPS activo. num_ltrp_entries[list_idx][rpl_sidx] especifica el número de entradas STRP en la estructura de sintaxis ref_pic_list_struct(list_idx, rpl_sidx).

- 30 delta_poc_st[list_idx][rpl_sidx][i], cuando la i-ésima entrada es la primera entrada STRP en la estructura de sintaxis ref_pic_list_struct(list_idx, rpl_sidx), especifica la diferencia entre los valores de recuento de orden de imágenes de la imagen actual y la imagen a la que hace referencia la i-ésima entrada o, cuando la i-ésima entrada es una entrada STRP, pero no la primera entrada STRP en la estructura de sintaxis ref_pic_list_struct(list_idx, rpl_sidx), especifica la diferencia entre los valores de recuento de orden de imágenes de las imágenes a las que hacen referencia la i-ésima entrada y la entrada STRP anterior en la estructura de sintaxis ref_pic_list_struct(list_idx, rpl_sidx). El valor de delta_poc_st[list_idx][rpl_sidx][i] estará en el intervalo de -215 a 215 - 1, inclusive.

Semántica de estructura de lista de imágenes de referencia a largo plazo.

- 40 La estructura de sintaxis ref_pic_list_ltrp_struct(ltrp_sidx) está presente en un SPS o en un encabezado de porción. Dependiendo de si la estructura de sintaxis está incluida en un encabezado de porción o en un SPS, se aplica lo siguiente: si está presente en un encabezado de porción, la estructura de sintaxis ref_pic_list_ltrp_struct(ltrp_sidx) especifica una lista de imágenes de referencia a largo plazo de la imagen actual (la imagen que contiene la porción). De lo contrario (presente en un SPS), la estructura de sintaxis ref_pic_list_ltrp_struct(list_idx, rpl_sidx) especifica un candidato para la lista de imágenes de referencia a largo plazo, y la expresión "la imagen actual" en la semántica especificada en el resto de esta sección se refiere a cada imagen que 1) tenga una o más porciones que contengan ref_pic_list_ltrp_idx[i] igual a un índice en la lista de las estructuras de sintaxis ref_pic_list_ltrp_struct(ltrp_sidx) incluidas en el SPS, y 2) esté en una CVS que tenga el SPS como SPS activo. num_ltrp_entries[ltrp_sidx] especifica el número de entradas LTRP en la estructura de sintaxis ref_pic_list_ltrp_struct(ltrp_sidx). poc_lsb_ltrp[rpl_sidx][i] especifica el valor del módulo de recuento de orden de imágenes MaxLtPicOrderCntLsb de la imagen a la que hace referencia la i-ésima entrada en la estructura de sintaxis ref_pic_list_ltrp_struct(rpl_sidx). La longitud del elemento de sintaxis poc_lsb_ltrp[rpl_sidx][i] es $\log_2(\text{MaxLtPicOrderCntLsb})$ bits. ltrp_pos_idx[rpl_sidx][i] especifica el índice de la i-ésima entrada en la estructura de sintaxis ref_pic_list_ltrp_struct(rpl_sidx) en la lista de imágenes de referencia después de la construcción de la lista de imágenes de referencia. La longitud del elemento de sintaxis ltrp_pos_idx[rpl_sidx][i] es $\log_2(\text{sps_max_dec_pic_buffering_minus1} + 1)$ bits. Cuando num_ltrp_entries[ltrp_sidx] es mayor de 1, poc_lsb_ltrp[rpl_sidx][i] y ltrp_pos_idx[rpl_sidx][i] estarán en el orden descendente de los valores ltrp_pos_idx[rpl_sidx][i].

Se describe el proceso de decodificación.

Proceso de decodificación para la construcción de listas de imágenes de referencia.

Este proceso se invoca al principio del proceso de decodificación para cada porción de una imagen no-IRAP. Las imágenes de referencia se direccionan a través de índices de referencia. Un índice de referencia es un índice en una lista de imágenes de referencia. Al decodificar una porción I, no se utiliza ninguna lista de imágenes de referencia en la decodificación de los datos de la porción. Al decodificar una porción P, sólo se utiliza la lista 0 de imágenes de referencia (es decir, RefPicList[0]) en la decodificación de los datos de la porción. Al decodificar una porción B, tanto la lista 0 de imágenes de referencia como la lista 1 de imágenes de referencia (es decir, RefPicList[1]) se utilizan en la decodificación de los datos de la porción. Al comienzo del proceso de decodificación para cada porción de una imagen no-IRAP, se obtienen las listas RefPicList[0] y RefPicList[1] de imágenes de referencia. Las listas de imágenes de referencia se utilizan en el marcado de imágenes de referencia o en la decodificación de los datos de la porción. Para una porción I de una imagen no-IRAP que no sea la primera porción de la imagen, RefPicList[0] y RefPicList[1] se obtienen con el propósito de verificar la conformidad del flujo de bits, pero su obtención no es necesaria para decodificar la o las imágenes actuales que siguen a la imagen actual en orden de decodificación. Para una porción P que no sea la primera porción de una imagen, RefPicList[1] se obtiene con el propósito de verificar la conformidad del flujo de bits, pero su obtención no es necesaria para decodificar la o las imágenes actuales que siguen a la imagen actual en orden de decodificación. Las listas RefPicList[0] y RefPicList[1] de imágenes de referencia se construyen de la siguiente manera:

```

for( i = 0; i < 2; i++ ) {
    if( ref_pic_list_sps_flag[ i ] )
        RplIdx[ i ] = ref_pic_list_idx[ i ]
    else
        RplIdx[ i ] = num_ref_pic_lists_in_sps[ i ]
    for( j = 0; posBase = PicOrderCntVal; j < NumEntriesInList[ i ][ RplIdx[ i ] ] ) {

        RefPicPoolList[ i ][ j ] = posBase + delta_pos[ i ][ RplIdx[ i ] ][ j ]
        if( hay una imagen picA de referencia en el DPS con PicOrderCntVal igual a RefPicPoolList[ i ][ j ] )
            RefPicList[ i ][ j ] = picA
        else
            RefPicList[ i ][ j ] = "ninguna imagen de referencia"
        posBase = RefPicPoolList[ i ][ j ]
    }

    if( ref_pic_list_lt_sps_flag[ i ] )
        LtRplIdx = ref_pic_list_lt_idx[ i ]
    else
        LtRplIdx = num_ref_pic_lists_lt_in_sps[ i ]
    for( j = 0; j < num_entries[ LtRplIdx[ i ] ][ j ] ) {
        if( hay una picA de referencia en el DPS con PicOrderCntVal < ( MaxLTPicOrderCntLst - 1 ) {
            igual a pos_idx[ LtRplIdx[ i ] ][ j ]
            for( k = sps_max_delta_pic_buffering_minus1; k > R_pos_idx[ LtRplIdx[ i ] ][ j ]; k-- )
                RefPicList[ i ][ k ] = RefPicList[ i ][ k - 1 ]
            RefPicList[ i ][ R_pos_idx[ LtRplIdx[ i ] ][ j ] ] = picA
        } else {
            for( k = sps_max_delta_pic_buffering_minus1; k > R_pos_idx[ LtRplIdx[ i ] ][ j ]; k-- )
                RefPicList[ i ][ k ] = RefPicList[ i ][ k - 1 ]
            RefPicList[ i ][ R_pos_idx[ LtRplIdx[ i ] ][ j ] ] = "ninguna imagen de referencia"
        }
    }
}

```

Para cada i igual a 0 o 1, se aplica lo siguiente: las primeras entradas NumRefIdxActive[i] en RefPicList[i] se denominan entradas activas en RefPicList[i], y las otras entradas en RefPicList[i] se denominan entradas inactivas en RefPicList[i]. Cada entrada en RefPicList[i][j] para j en el intervalo de 0 a NumEntriesInList[i][RplIdx[i]] - 1, inclusive, se conoce como una entrada STRP si lt_ref_pic_flag[i][RplIdx[i]][j] es igual a 0, y como una entrada LTRP de lo contrario. Es posible que a una imagen en particular hagan referencia tanto una entrada en RefPicList[0] como una entrada en RefPicList[1]. También es posible que a una imagen en particular hagan referencia más de una entrada en RefPicList[0] o más de una entrada en RefPicList[1]. Las entradas activas en RefPicList[0] y las entradas activas en RefPicList[1] se refieren colectivamente a todas las imágenes de referencia que se utilizan para la interpredicción de la imagen actual y una o más imágenes que siguen a la imagen actual en orden de decodificación. Las entradas inactivas en RefPicList[0] y las entradas inactivas en RefPicList[1] se refieren colectivamente a todas las imágenes de referencia que no se utilizan para la interpredicción de la imagen actual, pero se utilizan para la interpredicción de una o más imágenes que siguen a la imagen actual en orden de decodificación. Hay una o más entradas en RefPicList[0] o RefPicList[1] que son iguales a "ninguna imagen de referencia" porque las imágenes

correspondientes no están presentes en el DPB. Cada entrada inactiva en RefPicList[0] o RefPicList[0] que sea igual a "ninguna imagen de referencia" debe ignorarse. Se debe deducir una pérdida de imagen no intencionada para cada entrada activa en RefPicList[0] o RefPicList[1] que sea igual a "ninguna imagen de referencia".

- 5 Es un requisito de conformidad de flujo de bits que se apliquen las siguientes restricciones: para cada i igual a 0 o 1, el número de entradas en RefPicList[i] no será menor que NumRefIdxActive[i]. La imagen a la que haga referencia cada entrada activa en RefPicList[0] o RefPicList[1] estará presente en el DPB y tendrá TemporalId menor o igual que el de la imagen actual. Opcionalmente, se especifica además la siguiente restricción: el índice de entrada de cualquier entrada inactiva en RefPicList[0] o RefPicList[1] no se utilizará como índice de referencia para la decodificación de la imagen actual. Opcionalmente, se especifica además la siguiente restricción: una entrada inactiva en RefPicList[0] o RefPicList[1] no se referirá a la misma imagen que cualquier otra entrada en RefPicList[0] o RefPicList[1]. Una entrada STRP en RefPicList[0] o RefPicList[1] de una porción de una imagen y una entrada LTRP en RefPicList[0] o RefPicList[1] de la misma porción o de una porción diferente de la misma imagen no se referirán a la misma imagen. A la imagen actual en sí misma no hará referencia ninguna entrada en RefPicList[0] o RefPicList[1]. No habrá ninguna entrada LTRP en RefPicList[0] o RefPicList[1] para la que la diferencia entre PicOrderCntVal de la imagen actual y PicOrderCntVal de la imagen a la que hace referencia la entrada sea superior o igual a 224. Sea setOfRefPics el conjunto de imágenes únicas a las que hacen referencia todas las entradas de RefPicList[0] y todas las entradas de RefPicList[1]. El número de imágenes en setOfRefPics será menor o igual que sps_max_dec_pic_buffering_minus1, y setOfRefPics será el mismo para todas las porciones de una imagen.

Se expone la señalización del número de entradas de imágenes de referencia a corto plazo.

- En una realización alternativa de la divulgación, el elemento de sintaxis que especifica el número de entradas asociadas con imágenes de referencia a corto plazo en una estructura de lista de imágenes de referencia ref_pic_list_struct() se define como num_strp_entries_minus1[listIdx][rplIdx], en lugar de num_strp_entries[listIdx][rplIdx]. El cambio tiene dos efectos para la señalización de la lista de imágenes de referencia: guarda bits para señalar el número de entradas asociadas con la imagen de referencia a corto plazo en la estructura de lista de imágenes de referencia a medida que el elemento se codifica utilizando ue(v). Impone implícitamente una restricción tal que cada lista de imágenes de referencia contendrá al menos una imagen de referencia a corto plazo. Para acomodar esta idea se necesitan algunos cambios en relación con la primera realización.

- Para la señalización de listas de imágenes de referencia en encabezados de porción, sólo se señala la lista de imágenes de referencia necesaria de acuerdo con el tipo de porción, es decir, una lista de imágenes de referencia (es decir, la lista 0 de imágenes de referencia) para las porciones I o P y dos listas de imágenes de referencia (es decir, tanto la lista 0 de imágenes de referencia como la lista 1 de imágenes de referencia) para las porciones B. La sintaxis del encabezado de porción se cambia de la siguiente manera:

slice_header() {	Descriptor
slice_pic_parameter_set_id	ue(v)
slice_address	u(v)
slice_type	ue(v)
if (slice_type != I)	
log2_diff_ctu_max_bt_size	ue(v)
if(nal_unit_type != IRAP_NUT) {	
slice_pic_order_cnt_lsb	u(v)
for(i = 0; i < (slice_type == B ? 2 : 1); i++) {	
ref_pic_list_sps_flag[i]	u(1)
if(ref_pic_list_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1)	
ref_pic_list_idx[i]	u(v)
} else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i], long_term_ref_pics_flag)	
}	
if(slice_type == P slice_type == B) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag)	
for(i = 0; i < (slice_type == B ? 2 : 1); i++)	
num_ref_idx_active_minus1[i]	ue(v)
}	
}	
byte_alignment()	
}	

Aplicando el cambio anterior en el encabezado de porción (es decir, la lista 0 de imágenes de referencia para las

porciones I o P; la imagen 0 de referencia y la imagen 1 de referencia para las porciones B), se evitaría el esquema del problema en el que para una porción P sólo hay una imagen de referencia a corto plazo. Sin embargo, no se puede señalar una imagen de referencia a corto plazo duplicada en la lista 0 de imágenes de referencia y en la lista 1 de imágenes de referencia, cuando la entrada en la lista 1 de imágenes de referencia es una entrada inactiva, ya que el número de entradas activas en la lista 1 de imágenes de referencia debe ser igual a 0. La semántica de num_strp_entries_minus1[listIdx][rplIdx] se cambia de la siguiente manera: num_strp_entries_minus1[listIdx][rplIdx] plus 1 especifica el número de entradas STRP en la estructura de sintaxis ref_pic_list_struct(listIdx, rplIdx, ltrpFlag). La variable NumEntriesInList[listIdx][rplIdx] se obtiene de la siguiente manera: NumRefPicEntriesInRp1[listIdx][rplIdx] = num_strp_entries_minus1[listIdx][rplIdx] + 1 + num_ltrp_entries[listIdx][rplIdx]. El valor de NumRefPicEntries[listIdx][rplIdx] estará en el intervalo de 1 a sps_max_dec_pic_buffering_minus1, inclusive.

Permitir la inclusión de la imagen actual en listas de imágenes de referencia.

En una realización alternativa de la divulgación, se permite incluir la imagen actual en sus listas de imágenes de referencia. Para admitir esta característica, no se requiere ningún cambio de sintaxis y semántica en relación con esas descripciones en la primera y la segunda realizaciones. Sin embargo, las restricciones de conformidad de flujo de bits que se describen en el proceso de decodificación para la construcción de listas de imágenes de referencia tendrían que modificarse de la siguiente manera: es un requisito de conformidad de flujo de bits que se apliquen las siguientes restricciones: para cada i igual a 0 o 1, NumEntriesInList[i][RplIdx[i]] no será menor que NumRefIdxActive[i]. La imagen a la que haga referencia cada entrada activa en RefPicList[0] o RefPicList[1] estará presente en el DPB y tendrá TemporalId menor o igual que el de la imagen actual. Opcionalmente, se especifica además la siguiente restricción: el índice de entrada de cualquier entrada inactiva en RefPicList[0] o RefPicList[1] no se utilizará como índice de referencia para la decodificación de la imagen actual. Opcionalmente, se especifica además la siguiente restricción: una entrada inactiva en RefPicList[0] o RefPicList[1] no se referirá a la misma imagen que cualquier otra entrada en RefPicList[0] o RefPicList[1]. Una entrada STRP en RefPicList[0] o RefPicList[1] de una porción de una imagen y una entrada LTRP en RefPicList[0] o RefPicList[1] de la misma porción o de una porción diferente de la misma imagen no se referirán a la misma imagen. Cuando una entrada en RefPicList[i] haga referencia a la imagen actual, para i igual a 0 o 1, el índice de entrada será menor que NumRefIdxActive[i]. No habrá ninguna entrada LTRP en RefPicList[0] o RefPicList[1] para la que la diferencia entre PicOrderCntVal de la imagen actual y PicOrderCntVal de la imagen a la que hace referencia la entrada sea superior o igual a 224. Sea setOfRefPics el conjunto de imágenes únicas a las que hacen referencia todas las entradas de RefPicList[0] y todas las entradas de RefPicList[1]. Si la imagen actual no se incluye en setOfRefPics, el número de imágenes en setOfRefPics será menor o igual que sps_max_dec_pic_buffering_minus1; de lo contrario, el número de imágenes en setOfRefPics será menor o igual que sps_max_dec_pic_buffering_minus1 + 1. El setOfRefPics será el mismo para todas las porciones de una imagen.

Uso de diferentes bits LSB de POC para entradas LTRP en listas de imágenes de referencia.

En una realización alternativa de la divulgación, se permite que el número de bits utilizados para identificar imágenes de referencia a largo plazo en una estructura de lista de imágenes de referencia sea diferente entre la lista 0 de imágenes de referencia y la lista 1 de imágenes de referencia. Para admitir esta característica, se necesitan los siguientes cambios:

seq_parameter_set_rbsp() {	Descriptor
sps_seq_parameter_set_id	ue(v)
chroma_format_idc	ue(v)
if(chroma_format_idc == 3)	
separate_colour_plane_flag	u(1)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
bit_depth_luma_minus8	ue(v)
bit_depth_chroma_minus8	ue(v)
log2_max_pic_order_cnt_lsb_minus4	ue(v)
sps_max_dec_pic_buffering_minus1	ue(v)
qtbtt_dual_tree_intra_flag	ue(v)
log2_ctu_size_minus2	ue(v)
log2_min_qt_size_intra_slices_minus2	ue(v)
log2_min_qt_size_inter_slices_minus2	ue(v)
max_mtt_hierarchy_depth_inter_slices	ue(v)
max_mtt_hierarchy_depth_intra_slices	ue(v)
long_term_ref_pics_flag	u(1)
for($i = 0$; $i < 2$; $i++$) {	
if(long_term_ref_pics_flag)	
additional_lt_poc_lsb[i]	ue(v)
num_ref_pic_lists_in_sps[i]	ue(v)

for(j = 0; j < num_ref_pic_lists_in_sps[i]; j++)	
ref_pic_list_struct(i, j, long_term_ref_pics_flag)	
}	
rbs_trailing_bits()	
}	

Un additional_lt_poc_lsb[i] especifica el valor de la variable MaxLtPicOrderCntLsb[i] que se utiliza en el proceso de decodificación para la lista listIdx de imágenes de referencia igual a i de la siguiente manera: MaxLtPicOrderCntLsb[i] = 2(log2_max_pic_order_cnt_lsb_minus4 + 4 + additional_lt_poc_lsb[i]). El valor de additional_lt_poc_lsb[i] estará en el intervalo de 0 a 32 - log2_max_pic_order_cnt_lsb_minus4 - 4, inclusive. Cuando no está presente, se deduce que el valor de additional_lt_poc_lsb[i] es igual a 0.

Un poc_lsb_lt[listIdx][rplIdx][i] especifica el valor del módulo de recuento de orden de imágenes MaxLtPicOrderCntLsb[listIdx] de la imagen a la que hace referencia la i-ésima entrada en la estructura de sintaxis ref_pic_list_struct(listIdx, rplIdx, ltrpFlag). La longitud del elemento de sintaxis poc_lsb_lt[listIdx][rplIdx][i] es Log2(MaxLtPicOrderCntLsb[listIdx]) bits.

Las listas RefPicList[0] y RefPicList[1] de imágenes de referencia se construyen de la siguiente manera:

```

for( i = 0; i < 2; i++ ) {
    if( ref_pic_list_sps_flag[ i ] )
        RplIdx[ i ] = ref_pic_list_idx[ i ];
    else
        RplIdx[ i ] = num_ref_pic_lists_in_sps[ i ];
    for( j = 0; pocBase = PicOrderCntVal, j < NumEntriesInList[ i ][ RplIdx[ i ] ]; j++ ) {
        if( ! ref_pic_flag[ i ][ RplIdx[ i ] ][ j ] )
            RefPicPoolList[ i ][ j ] = pocBase + delta_pic_order[ i ][ RplIdx[ i ] ][ j ];
        if( hay una imagen picA de referencia en el DPB con PicOrderCntVal igual a RefPicPoolList[ i ][ j ] )
            RefPicList[ i ][ j ] = picA;
        else
            RefPicList[ i ][ j ] = "ninguna imagen de referencia";
        pocBase = RefPicPoolList[ i ][ j ];
    }
    if( hay una picA de referencia en el DPB con PicOrderCntVal & ( MaxLtPicOrderCntLsb[ i ] - 1 )
        igual a poc_lsb_lt[ i ][ RplIdx[ i ] ][ j ] )
        RefPicList[ i ][ j ] = picA;
    else
        RefPicList[ i ][ j ] = "ninguna imagen de referencia";
}
}

```

Utilización del mismo ref_pic_list_sps_flag para las listas 0 y 1 de imágenes de referencia.

En una realización alternativa de la divulgación, en lugar de utilizar dos señalizadores para indicar si la lista 0 de imágenes de referencia y la lista 1 de imágenes de referencia se obtienen sobre la base de estructuras de sintaxis ref_pic_list_struct() en el SPS activo, se utiliza un señalizador para ambas listas de imágenes de referencia. Tal alternativa restringe que, bien ambas listas de imágenes de referencia se obtengan sobre la base de ref_pic_list_struct() en el SPS activo, bien se obtengan sobre la base de estructuras de sintaxis ref_pic_list_struct() que se incluyen directamente en los encabezados de porción de la imagen actual. Para admitir esta característica, se necesitan los siguientes cambios:

slice_header() {	Descriptor
slice_pic_parameter_set_id	ue(v)
slice_address	u(v)
slice_type	ue(v)
if(slice_type != I)	
log2_diff_ctu_max_bt_size	ue(v)
if(nal_unit_type != IRAP_NUT) {	
slice_pic_order_cnt_lsb	u(v)
ref_pic_list_sps_flag	u(1)
for(i = 0; i < 2; i++) {	
if(ref_pic_list_sps_flag) {	
if(num_ref_pic_lists_in_sps[i] > 1)	
ref_pic_list_idx[i]	u(v)
} else	

ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i], long_term_ref_pics_flag)	
}	
if(slice_type == P slice_type == B) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag)	
for(i = 0; i < (slice_type == B ? 2 : 1); i++)	
num_ref_idx_active_minus1[i]	ue(v)
}	
}	
byte_alignment()	
}	

Un ref_pic_list_sps_flag igual a 1 especifica que las listas de imágenes de referencia de la imagen actual se obtienen sobre la base de las estructuras de sintaxis ref_pic_list_struct(listIdx, rplIdx, ltrpFlag) en el SPS activo. ref_pic_list_sps_flag igual a 0 especifica que las listas de imágenes de referencia de la imagen actual se obtienen sobre la base de las estructuras de sintaxis ref_pic_list_struct(listIdx, rplIdx, ltrpFlag) que se incluyen directamente en los encabezados de porción de la imagen actual. Cuando num_ref_pic_lists_in_sps[0] o num_ref_pic_lists_in_sps[1] son iguales a 0, el valor de ref_pic_list_sps_flag será igual a 0. pic_lists_in_sps[1] es igual a 0, el valor de ref_pic_list_sps_flag será igual a 0.

- 10 Las listas RefPicList[0] y RefPicList[1] de imágenes de referencia se construyen de la siguiente manera:

```
for( i = 0; i < 2; i++ ) {
    if( ref_pic_list_sps_flag == 1 )
        RplIdx[ i ] = ref_pic_list_idx[ i ]
    else
        RplIdx[ i ] = num_ref_pic_lists_in_sps[ i ]
    for( j = 0; pocBase = PicOrderCntVal; j < NumEntriesInList[ i ][ RplIdx[ i ][ j ] ] ) {
        if( ! ( ref_pic_flag[ i ][ RplIdx[ i ][ j ] ] ) ) {
            RefPicPosList[ i ][ j ] = pocBase - delta_poc[ i ][ RplIdx[ i ][ j ] ]
            if( hay una imagen picA de referencia en el DFB con PicOrderCntVal igual a RefPicPosList[ i ][ j ] )
                RefPicList[ i ][ j ] = picA
            else
                RefPicList[ i ][ j ] = "ninguna imagen de referencia"
            pocBase = RefPicPosList[ i ][ j ]
        } else {
            if( hay una picA de referencia en el DFB con PicOrderCntVal & { MaxLtrPicOrderCntLsb - 1 }
                igual a poc_lsb[ i ][ RplIdx[ i ][ j ] ] )
                RefPicList[ i ][ j ] = picA
            else
                RefPicList[ i ][ j ] = "ninguna imagen de referencia"
        }
    }
}
```

- 15 Señalización del Bit Más Significativo (MSB, por sus siglas en inglés) de delta POC para entradas de imagen de referencia a largo plazo.

- 20 En una realización alternativa de la divulgación, en lugar de usar bits adicionales para representar LSB de POC de entradas de imagen de referencia a largo plazo en ref_pic_list_struct(), se señala el ciclo de MSB de POC para diferenciar imágenes de referencia a largo plazo. Cuando se señala, la información del ciclo de MSB de POC se señala para cada entrada en ref_pic_list_struct() que haga referencia a una imagen de referencia a largo plazo. La estructura de sintaxis ref_pic_list_struct() no está señalizada en SPS, sino sólo en encabezados de porción. Para admitir esta característica, se necesitan los siguientes cambios:

seq_parameter_set_rbsp() {	Descriptor
sps_seq_parameter_set_id	ue(v)
chroma_format_idc	ue(v)
if(chroma_format_idc == 3)	
separate_colour_plane_flag	u(1)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
bit_depth_luma_minus8	ue(v)

bit_depth_chroma_minus8	ue(v)
log2_max_pic_order_cnt_lsb_minus4	ue(v)
sps_max_dec_pic_buffering_minus1	ue(v)
qtbtt_dual_tree_intra_flag	ue(v)
log2_ctu_size_minus2	ue(v)
log2_min_qt_size_intra_slices_minus2	ue(v)
log2_min_qt_size_inter_slices_minus2	ue(v)
max_mtt_hierarchy_depth_inter_slices	ue(v)
max_mtt_hierarchy_depth_intra_slices	ue(v)
long_term_ref_pics_flag	u(1)
rbsp_trailing_bits()	
}	

slice_header() {	Descriptor
slice_pic_parameter_set_id	ue(v)
slice_address	u(v)
slice_type	ue(v)
if (slice_type != I)	
log2_diff_ctu_max_bt_size	ue(v)
if(nal_unit_type != IRAP_NUT) {	
slice_pic_order_cnt_lsb	u(v)
for(i = 0; i < 2; i++) {	
ref_pic_list_struct(i, long_term_ref_pics_flag)	
}	
if(slice_type == P slice_type == B) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag)	
for(i = 0; i < (slice_type == B ? 2 : 1); i++)	
num_ref_idx_active_minus1[i]	ue(v)
}	
}	
byte_alignment()	
}	

ref_pic_list_struct(listIdx, ltrpFlag) {	Descriptor
num_strp_entries[listIdx]	ue(v)
if(ltrpFlag)	
num_ltrp_entries[listIdx]	ue(v)
for(i = 0; i < NumEntriesInList[listIdx]; i++) {	
if(num_ltrp_entries[listIdx] > 0)	
lt_ref_pic_flag[listIdx][i]	
if(!lt_ref_pic_flag[listIdx][i])	
delta_poc_st[listIdx][i]	se(v)
else {	
poc_lsb_lt[listIdx][i]	u(v)
delta_poc_msb_present_flag[listIdx][i]	u(1)
if(delta_poc_msb_present_flag[listIdx][i])	
delta_poc_msb_cycle_lt[listIdx][i]	ue(v)
}	
}	
}	
}	

La estructura de sintaxis ref_pic_list_struct(listIdx, ltrpFlag) está presente en un encabezado de porción. Cuando está presente en un encabezado de porción, la estructura de sintaxis ref_pic_list_struct(listIdx, ltrpFlag) especifica la lista listIdx de imágenes de referencia de la imagen actual (la imagen que contiene la porción). num_strp_entries[listIdx] especifica el número de entradas STRP en la estructura de sintaxis ref_pic_list_struct(listIdx, ltrpFlag). num_ltrp_entries[listIdx] especifica el número de entradas LTRP en la estructura de sintaxis ref_pic_list_struct(listIdx, ltrpFlag). Cuando no está presente, se deduce que el valor de num_ltrp_entries[listIdx][rplIdx] es igual a 0.

La variable NumEntriesInList[listIdx] se obtiene de la siguiente manera: NumRefPicEntriesInRpl[listIdx] = num_strp_entries[listIdx] num_ltrp_entries[listIdx]

El valor de NumRefPicEntries[listIdx] estará en el intervalo de 0 a sps_max_dec_pic_buffering_minus1, inclusive.

It_ref_pic_flag[listIdx][i] igual a 1 especifica que la i-ésima entrada en la estructura de sintaxis ref_pic_list_struct(listIdx, ltrpFlag) es una entrada LTRP. It_ref_pic_flag[listIdx][i] igual a 0 especifica que la i-ésima entrada en la estructura de sintaxis ref_pic_list_struct(listIdx, ltrpFlag) es una entrada STRP. Cuando no está presente, se deduce que el valor de It_ref_pic_flag[listIdx][i] es igual a 0. Es un requisito de conformidad de flujo de bits que la suma de It_ref_pic_flag[listIdx][i] para todos los valores de i en el intervalo de 0 a NumRefPicEntries[listIdx] - 1, inclusive, sea igual a num_ltrp_entries[listIdx]. delta_poc_st[listIdx][i], cuando la i-ésima entrada es la primera entrada STRP en la estructura de sintaxis ref_pic_list_struct(listIdx, ltrpFlag), especifica la diferencia entre los valores de recuento de orden de imágenes de la imagen actual y la imagen a la que hace referencia la i-ésima entrada o, cuando la i-ésima entrada es una entrada STRP, pero no la primera entrada STRP, en la estructura de sintaxis ref_pic_list_struct(listIdx, ltrpFlag), especifica la diferencia entre los valores de recuento de orden de imágenes de las imágenes a las que hacen referencia la i-ésima entrada y la entrada STRP anterior en la estructura de sintaxis ref_pic_list_struct(listIdx, ltrpFlag). El valor de delta_poc_st[listIdx][i] estará en el intervalo de -2^{15} a $2^{15} - 1$, inclusive. poc_lsb_lt[listIdx][i] especifica el valor del módulo de recuento de orden de imágenes MaxLtPicOrderCntLsb de la imagen a la que hace referencia la i-ésima entrada en la estructura de sintaxis ref_pic_list_struct(listIdx, ltrpFlag). La longitud del elemento de sintaxis poc_lsb_lt[listIdx][i] es $\text{Log2}(\text{MaxPicOrderCntLsb})$ bits. delta_poc_msb_present_flag[listIdx][i] igual a 1 especifica que delta_pocmsb_cycle_lt[listIdx][i] está presente. delta_poc_msb_present_flag[listIdx][i] igual a 0 especifica que delta_poc_msb_cycle_lt[listIdx][i] no está presente. Cuando num_ltrp_entries[listIdx] sea mayor de 0 y haya más de una imagen de referencia en el DPB en el momento en que se decodifica este encabezado de porción para la que el módulo MaxPicOrderCntLsb de PicOrderCntVal sea igual a poc_lsb_lt[listIdx][i], delta_poc_msb_present_flag[listIdx][i] será igual a 1. Cuando no está presente, se deduce que el valor de delta_poc_msb_cycle_lt[listIdx][i] es igual a 0. delta_pocmsb_cycle_lt[listIdx][i] se utiliza para determinar el valor de los bits más significativos del valor de recuento de orden de imágenes de la i-ésima entrada en la estructura de sintaxis ref_pic_list_struct(listIdx, ltrpFlag). Cuando delta_poc_msb_cycle_lt[listIdx][i] no está presente, se deduce que es igual a 0. Cambios en el proceso de decodificación para el recuento de orden de imágenes:

Las listas RefPicList[0] y RefPicList[1] de imágenes de referencia se construyen de la siguiente manera:

```

for( i = 0; i < 2; i++ ) {
    for( j = 0, pocBase = PicOrderCntVal; j < NumEntriesInList[ i ]; j++ ) {
        if( It_ref_pic_flag[ i ][ j ] ) {
            RefPicPocList[ i ][ j ] = pocBase + delta_poc_st[ i ][ j ];
            if( hay una imagen picA de referencia en el DPB con PicOrderCntVal igual a RefPicPocList[ i ][ j ] )
                RefPicList[ i ][ j ] = picA;
            else
                RefPicList[ i ][ j ] = "ninguna imagen de referencia";
            pocBase = RefPicPocList[ i ][ j ];
        } else {
            if( delta_poc_msb_cycle_lt[ i ][ j ] es igual a 0 &&
                hay una imagen picA de referencia en el DPB con PicOrderCntVal &
                { MaxLtPicOrderCntLsb - 1 } igual a poc_lsb_lt[ i ][ j ] )
                RefPicList[ i ][ j ] = picA;
            else if( delta_poc_msb_cycle_lt[ i ][ j ] es igual a 1 &&
                hay una imagen picA de referencia en el DPB con PicOrderCntVal igual a
                { MaxPicOrderCntLsb * delta_poc_msb_cycle_lt[ i ][ j ] + poc_lsb_lt[ i ][ j ] } )
                RefPicList[ i ][ j ] = picA;
            else
                RefPicList[ i ][ j ] = "ninguna imagen de referencia";
        }
    }
}

```

Como alternativa, la semántica de delta_poc_msb_cycle_lt[listIdx][i] puede expresarse como delta de delta, de manera que la construcción de listas de imágenes de referencia se puede actualizar de la siguiente manera: las listas RefPicList[0] y RefPicList[1] de imágenes de referencia se construyen de la siguiente manera:

```

for( j = 0; j < 2; j++ ) {

    prevMsbCycle = 0;
    for( j = 0; picBase = PicOrderCntVal; j < NumEntriesInList[ i ] ) {
        if( ! ref_pic_flag[ i ][ j ] ) {
            RefPicList[ i ][ j ] = picBase + delta_poc_at[ i ][ j ];
            if( hay una imagen picA de referencia en el DPB con PicOrderCntVal igual a RefPicList[ i ][ j ] )
                RefPicList[ i ][ j ] = picA;
            else
                RefPicList[ i ][ j ] = "ninguna imagen de referencia";
            picBase = RefPicList[ i ][ j ];
        } else {
            if( delta_poc_msb_cycle_R[ i ][ j ] es igual a 0 &&
                hay una imagen picA de referencia en el DPB con PicOrderCntVal &
                ( MaxPicOrderCntLsb - 1 ) igual a poc_lsb_R[ i ][ j ] )
                RefPicList[ i ][ j ] = picA;
            else if( delta_poc_msb_cycle_R[ i ][ j ] es igual a 1 &&
                hay una picA de referencia en el DPB con PicOrderCntVal igual a
                ( MaxPicOrderCntLsb * ( delta_poc_msb_cycle_R[ i ][ j ] +
                    prevMsbCycle )
                    + poc_lsb_R[ i ][ j ] ) )
                RefPicList[ i ][ j ] = picA;
            prevMsbCycle += delta_poc_msb_cycle_R[ i ][ j ];
        } else
            RefPicList[ i ][ j ] = "ninguna imagen de referencia";
    }
}

```

Es un requisito de conformidad de flujo de bits que se apliquen las siguientes restricciones: para cada i igual a 0 o 1, NumEntriesInList[i][RplIdx[i]] no será menor que NumRefIdxActive[i]. La imagen a la que haga referencia cada entrada activa en RefPicList[0] o RefPicList[1] estará presente en el DPB y tendrá TemporalId menor o igual que el de la imagen actual. Opcionalmente, se especifica además la siguiente restricción: el índice de entrada de cualquier entrada inactiva en RefPicList[0] o RefPicList[1] no se utilizará como índice de referencia para la decodificación de la imagen actual. Opcionalmente, se especifica además la siguiente restricción: una entrada inactiva en RefPicList[0] o RefPicList[1] no se referirá a la misma imagen que cualquier otra entrada en RefPicList[0] o RefPicList[1]. Una entrada STRP en RefPicList[0] o RefPicList[1] de una porción de una imagen y una entrada LTRP en RefPicList[0] o RefPicList[1] de la misma porción o de una porción diferente de la misma imagen no se referirán a la misma imagen. A la imagen actual en sí misma no hará referencia ninguna entrada en RefPicList[0] o RefPicList[1]. No habrá ninguna entrada LTRP en RefPicList[0] o RefPicList[1] para la que la diferencia entre PicOrderCntVal de la imagen actual y PicOrderCntVal de la imagen a la que hace referencia la entrada sea superior o igual a 224. Sea setOfRefPics el conjunto de imágenes únicas a las que hacen referencia todas las entradas de RefPicList[0] y todas las entradas de RefPicList[1]. El número de imágenes en setOfRefPics será menor o igual que sps_max_dec_pic_buffering_minus1, y setOfRefPics será el mismo para todas las porciones de una imagen.

Cada STRP se identifica por su valor PicOrderCntVal. Para cada LTRP, si hace referencia a la misma una entrada en RefPicList[0] o RefPicList[1] con delta_poc_msb_present_flag[listIdx][i] igual a 1, se identifica por su valor PicOrderCntVal, de lo contrario, se identifica por Log2(MaxPicOrderCntLsb) LSB de su valor PicOrderCntVal.

Alternativa 1 de señalización de MSB de delta POC para las entradas de imágenes de referencia a largo plazo.

Esta realización proporciona una alternativa a la realización descrita en la sección anterior. De manera similar a la idea de la sección anterior, en lugar de usar bits adicionales para representar LSB de POC de imagen de referencia a largo plazo en ref_pic_list_struct(), se señala el ciclo de MSB de POC para diferenciar imágenes de referencia a largo plazo. Sin embargo, en esta alternativa, cuando se señala, la información del ciclo de MSB de POC no se señala dentro de ref_pic_list_struct(), en cambio, cuando se necesita información del ciclo de MSB de POC, se señala en el encabezado de porción. La estructura de sintaxis ref_pic_list_struct() se señala en el SPS y en encabezados de porción.

seq_parameter_set_rbsp() {	Descriptor
sps_seq_parameter_set_id	ue(v)
chroma_format_idc	ue(v)
if(chroma_format_idc == 3)	
separate_colour_plane_flag	u(1)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)

bit_depth_luma_minus8	ue(v)
bit_depth_chroma_minus8	ue(v)
log2_max_pic_order_cnt_lsb_minus4	ue(v)
sps_max_dec_pic_buffering_minus1	ue(v)
qtbtt_dual_tree_intra_flag	ue(v)
log2_ctu_size_minus2	ue(v)
log2_min_qt_size_intra_slices_minus2	ue(v)
log2_min_qt_size_inter_slices_minus2	ue(v)
max_mtt_hierarchy_depth_inter_slices	ue(v)
max_mtt_hierarchy_depth_intra_slices	ue(v)
long_term_ref_pics_flag	u(1)
for(i = 0; i < 2; i++) {	
num_ref_pic_lists_in_sps[i]	ue(v)
for(j = 0; j < num_ref_pic_lists_in_sps[i]; j++)	
ref_pic_list_struct(i, j, long_term_ref_pics_flag)	
}	
rbsp_trailing_bits()	
}	

slice_header() {	Descriptor
slice_pic_parameter_set_id	ue(v)
slice_address	u(v)
slice_type	ue(v)
if(slice_type != I)	
log2_diff_ctu_max_bt_size	ue(v)
if(nal_unit_type != IRAP_NUT) {	
slice_pic_order_cnt_lsb	u(v)
for(i = 0; i < 2; i++) {	
ref_pic_list_sps_flag[i]	u(1)
if(ref_pic_list_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1)	
ref_pic_list_idx[i]	u(v)
} else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i], long_term_ref_pics_flag)	
if(long_term_ref_pics_flag) {	
NumLtrpEntries[i] = ref_pic_list_sps_flag[i] ?	
num_ltrp_entries[i][ref_pic_list_idx[i]] :	
num_ltrp_entries[i][num_ref_pic_lists_in_sps[i]]	
for(j = 0; j < NumLtrpEntries[i]; j++) {	
delta_poc_msb_present_flag[i][j]	u(1)
if(delta_poc_msb_present_flag[i][j])	
delta_poc_msb_cycle_ltrp[i][j]	ue(v)
}	
}	
}	
}	
if(slice_type == P slice_type == B) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag)	
for(i = 0; i < (slice_type == B ? 2 : 1); i++)	
num_ref_idx_active_minus1[i]	ue(v)
}	
}	
byte_alignment()	
}	

delta_pocmsb_present_flag[i][j] igual a 1 especifica que delta_poc_msb_cycle_ltrp[i][j] está presente. delta_poc_msb_present_flag[i][j] igual a 0 especifica que delta_poc_msb_cycle_ltrp[i][j] no está presente. Cuando NumLtrpEntries[i] sea mayor de 0 y para la j-ésima entrada LTRP en la estructura de sintaxis ref_pic_list_struct(i, rplIdx, 1) haya más de una imagen de referencia en el DPB en el momento en que se decodifica este encabezado de porción para las que el módulo MaxPicOrderCntLsb de PicOrderCntVal sea igual a poc_lsb_ltrp[i][rplIdx][j], donde jj es el índice de entrada de la entrada en la estructura de sintaxis ref_pic_list_struct(i, rplIdx, 1) que es la j-ésima entrada LTRP en la estructura de sintaxis ref_pic_list_struct(i, rplIdx, 1), delta_poc_msb_present_flag[i][j] será igual a 1. Cuando no está presente, se deduce que el valor de delta_poc_msb_cycle_ltrp[i][j] es igual a 0.

delta_poc_msb_cycle_lt[i][j] se utiliza para determinar el valor de los bits más significativos del valor de recuento de orden de imágenes de la j-ésima entrada LTRP en la estructura de sintaxis ref_pic_list_struct(i, rplIdx, 1). Cuando delta_poc_msb_cycle_lt[i][j] no está presente, se deduce que es igual a 0.

slice_header() {	Descriptor
slice_pic_parameter_set_id	ue(v)
slice_address	u(v)
slice_type	ue(v)
if (slice_type != I)	
log2_diff_ctu_max_bt_size	ue(v)
if(nal_unit_type != IRAP_NUT) {	
slice_pic_order_cnt_lsb	u(v)
for(i = 0; i < 2; i++) {	
ref_pic_list_sps_flag[i]	u(1)
if(ref_pic_list_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1)	
ref_pic_list_idx[i]	u(v)
} else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i], long_term_ref_pics_flag)	
if(long_term_ref_pics_flag) {	
rplIdx = ref_pic_list_sps_flag[i] ? ref_pic_list_idx[i] :	
num_ref_pic_lists_in_sps[i]	
numRplEntries[i] = num_strp_entries[i][rplIdx] +	
num_ltrp_entries[i][rplIdx]	
NumLtrpEntries[i] = num_ltrp_entries[i][rplIdx]	
for(j = 0; j < numRplEntries[i]; j++) {	
if(lt_ref_pic_flag[i][rplIdx][j])	
delta_poc_msb_present_flag[i][j]	u(1)
if(delta_poc_msb_present_flag[i][j])	
delta_poc_msb_cycle_lt[i][j]	ue(v)
}	
}	
}	
if(slice_type == P slice_type == B) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag)	
for(i = 0; i < (slice_type == B ? 2 : 1); i++)	
num_ref_idx_active_minus1[i]	ue(v)
}	
}	
byte_alignment()	
}	

5 delta_pocmsb_present_flag[i][j] igual a 1 especifica que delta_poc_msb_cycle_lt[i][j] está presente. delta_pocmsb_present_flag[i][j] igual a 0 especifica que delta_poc_msb_cycle_lt[i][j] no está presente. Cuando NumLtrpEntries[i] sea mayor de 0 y haya más de una imagen de referencia en el DPB en el momento en que se decodifica este encabezado de porción para las que el módulo MaxPicOrderCntLsb de PicOrderCntVal sea igual a

10 poc_lsb_lt[i][rplIdx][j], delta_pocmsb_present_flag[i][j] será igual a 1. Cuando no está presente, se deduce que el valor de delta_poc_msb_cycle_lt[i][j] es igual a 0. delta_poc_msb_cycle_lt[i][j] se utiliza para determinar el valor de los bits más significativos del valor de recuento de orden de imágenes de la j-ésima entrada en la estructura de sintaxis ref_pic_list_struct(i, rplIdx, 1). Cuando delta_pocmsb_cycle_lt[i][j] no está presente, se deduce que es igual a 0. poc_lsb_lt[listIdx][rplIdx][i] especifica el valor del módulo de recuento de orden de imágenes

15 MaxPicOrderCntLsb de la imagen a la que hace referencia la i-ésima entrada en la estructura de sintaxis ref_pic_list_struct(listIdx, rplIdx, ltrpFlag). La longitud del elemento de sintaxis poc_lsb_lt[listIdx][rplIdx][i] es Log2(MaxPicOrderCntLsb) bits.

Cambios en el proceso de decodificación para el recuento de orden de imágenes:

20 Para el diseño 1 de encabezado de porción, las listas RefPicList[0] y RefPicList[1] de imágenes de referencia se construyen de la siguiente manera:

```

for( i = 0; i < 2; i++) {
    if( ref_pic_list_sps_flag[ i ] )
        RplIdx[ i ] = ref_pic_list_idx[ i ]
    else
        RplIdx[ i ] = num_ref_pic_lists_in_sps[ i ]
    msbCycleIdx = 0

    for( j = 0; pocBase = PicOrderCntVal; j < NumEntriesInList[ i ][ RplIdx[ i ] ], j++) {
        if( !t_ref_pic_flag[ i ][ RplIdx[ i ] ][ j ] ) {
            RefPicPocList[ i ][ j ] = pocBase - delta_poc_st[ i ][ RplIdx[ i ] ][ j ]
            if( hay una imagen picA de referencia en el DPB con PicOrderCntVal igual a RefPicPocList[ i ][ j ] )
                RefPicList[ i ][ j ] = picA
            else
                RefPicList[ i ][ j ] = "ninguna imagen de referencia"
            pocBase = RefPicPocList[ i ][ j ]
        } else {
            if( delta_poc_msb_cycle_lt[ i ][ msbCycleIdx ] es igual a 0 &&
                hay una imagen picA de referencia en el DPB con PicOrderCntVal &
                ( MaxLtpicOrderCntLsb - 1 ) igual a poc_lsb_lt[ i ][ RplIdx[ i ] ][ j ] )
                RefPicList[ i ][ j ] = picA
            else if( delta_poc_msb_cycle_lt[ i ][ msbCycleIdx ] es igual a 1 &&
                hay una imagen picA de referencia en el DPB con PicOrderCntVal igual a
                ( MaxPicOrderCntLsb *
                delta_poc_msb_cycle_lt[ i ][ msbCycleIdx ] + poc_lsb_lt[ i ][ RplIdx[ i ] ][ j ] ) ) {
                RefPicList[ i ][ j ] = picA
                msbCycleIdx++
            } else
                RefPicList[ i ][ j ] = "ninguna imagen de referencia"
        }
    }
}

```

Como alternativa, para el diseño 1 de encabezado de porción, la semántica de delta_poc_msb_cycle_lt[listIdx][i] puede expresarse como delta de delta, de manera que la construcción de listas de imágenes de referencia se puede actualizar de la siguiente manera: las listas RefPicList[0] y RefPicList[1] de imágenes de referencia se construyen de la siguiente manera:

```

for( i = 0; i < 2; i++) {
    if( ref_pic_list_sps_flag[ i ] )
        RplIdx[ i ] = ref_pic_list_idx[ i ]
    else
        RplIdx[ i ] = num_ref_pic_lists_in_sps[ i ]
    prevMsbCycle = 0
    msbCycleIdx = 0
    for( j = 0; pocBase = PicOrderCntVal; j < NumEntriesInList[ i ][ RplIdx[ i ] ], j++) {
        if( !t_ref_pic_flag[ i ][ RplIdx[ i ] ][ j ] ) {
            RefPicPocList[ i ][ j ] = pocBase - delta_poc_st[ i ][ RplIdx[ i ] ][ j ]
            if( hay una imagen picA de referencia en el DPB con PicOrderCntVal igual a RefPicPocList[ i ][ j ] )
                RefPicList[ i ][ j ] = picA
            else
                RefPicList[ i ][ j ] = "ninguna imagen de referencia"
            pocBase = RefPicPocList[ i ][ j ]
        } else {
            if( delta_poc_msb_cycle_lt[ i ][ msbCycleIdx ] es igual a 0 &&
                hay una imagen picA de referencia en el DPB con PicOrderCntVal &
                ( MaxLtpicOrderCntLsb - 1 ) igual a poc_lsb_lt[ i ][ RplIdx[ i ] ][ j ] )
                RefPicList[ i ][ j ] = picA
            else if( delta_poc_msb_cycle_lt[ i ][ msbCycleIdx ] es igual a 1 &&
                hay una imagen picA de referencia en el DPB con PicOrderCntVal igual a
                ( MaxPicOrderCntLsb * ( delta_poc_msb_cycle_lt[ i ][ msbCycleIdx ] +
                prevMsbCycle ) + poc_lsb_lt[ i ][ RplIdx[ i ] ][ j ] ) ) {
                RefPicList[ i ][ j ] = picA
                prevMsbCycle = delta_poc_msb_cycle_lt[ i ][ msbCycleIdx ]
                msbCycleIdx++
            }
        }
    }
}

```

```

    } else
        RefPicList[i][j] = "ninguna imagen de referencia"
    }
}
}

```

Para el diseño 2 de encabezado de porción, las listas RefPicList[0] y RefPicList[1] de imágenes de referencia se construyen de la siguiente manera:

5

```

for( i = 0; i < 2; i++ ) {
    if( ref_pic_list_sps_flag[i] )
        RplIdx[i] = ref_pic_list_idx[i]
    else
        RplIdx[i] = nom_ref_pic_lists_in_sps[i]
    for( j = 0, pocBase = PicOrderCntVal; j < NumEntriesInList[i][ RplIdx[i] ][ j ]++ ) {
        if( ! ref_pic_flag[i][ RplIdx[i] ][ j ] ) {
            RefPicPoolList[i][ j ] = pocBase + delta_poc_at[i][ RplIdx[i] ][ j ]
            if( hay una imagen picA de referencia en el DPB con PicOrderCntVal igual a RefPicPoolList[i][ j ] )
                RefPicList[i][ j ] = picA
            else
                RefPicList[i][ j ] = "ninguna imagen de referencia"
            pocBase = RefPicPoolList[i][ j ]
        } else {
            if( delta_poc_msb_cycle_it[i][ j ] es igual a 0 &&
                hay una imagen picA de referencia en el DPB con PicOrderCntVal &
                ( MaxPicOrderCntLsb - 1 ) igual a poc_lsb_it[i][ RplIdx[i] ][ j ] )
                RefPicList[i][ j ] = picA
            else if( delta_poc_msb_cycle_it[i][ j ] es igual a 1 &&
                hay una imagen picA de referencia en el DPB con PicOrderCntVal igual a (
                MaxPicOrderCntLsb *
                delta_poc_msb_cycle_it[i][ j ] ) + poc_lsb_it[i][ RplIdx[i] ][ j ] )
                RefPicList[i][ j ] = picA
            else
                RefPicList[i][ j ] = "ninguna imagen de referencia"
        }
    }
}
}

```

Como alternativa, para el diseño 2 de encabezado de porción, la semántica de delta_poc_msb_cycle_it[listIdx][i] puede expresarse como delta de delta, de manera que la construcción de listas de imágenes de referencia se puede actualizar de la siguiente manera: las listas RefPicList[0] y RefPicList[1] de imágenes de referencia se construyen de la siguiente manera:

10

```

for( i = 0; i < 2; i++ ) {
    if( ref_pic_list_sps_flag[i] )
        RplIdx[i] = ref_pic_list_idx[i]
    else
        RplIdx[i] = nom_ref_pic_lists_in_sps[i]
    prevMsbCycle = 0
    for( j = 0, pocBase = PicOrderCntVal; j < NumEntriesInList[i][ RplIdx[i] ][ j ]++ ) {
        if( ! ref_pic_flag[i][ RplIdx[i] ][ j ] ) {
            RefPicPoolList[i][ j ] = pocBase + delta_poc_at[i][ RplIdx[i] ][ j ]
            if( hay una imagen picA de referencia en el DPB con PicOrderCntVal igual a RefPicPoolList[i][ j ] )
                RefPicList[i][ j ] = picA
            else
                RefPicList[i][ j ] = "ninguna imagen de referencia"
            pocBase = RefPicPoolList[i][ j ]
        } else {
            if( delta_poc_msb_cycle_it[i][ j ] es igual a 0 &&

```

```

        hay una imagen picA de referencia en el DPB con PicOrderCntVal &
        ( MaxPicOrderCntLsb - 1 ) igual a poc_lsb_R[ i ][ RplIdx[ i ][ j ] ] }
    RefPicList[ i ][ j ] = picA
else if( delta_poc_msb_cycle_R[ i ][ msbCycleIdx ] es igual a 1 &&
        hay una imagen picA de referencia en el DPB con PicOrderCntVal igual a
        ( MaxPicOrderCntLsb * ( delta_poc_msb_cycle_R[ i ][ j ] ) +
        prevMsbCycle ) + poc_lsb_R[ i ][ RplIdx[ i ][ j ] ] ) {
    RefPicList[ i ][ j ] = picA
    prevMsbCycle += delta_poc_msb_cycle_R[ i ][ j ]
} else
    RefPicList[ i ][ j ] = "ninguna imagen de referencia"
}
}
}

```

Es un requisito de conformidad de flujo de bits que se apliquen las siguientes restricciones: para cada i igual a 0 o 1, $\text{NumEntriesInList}[i][\text{RplIdx}[i][j]]$ no será menor que $\text{NumRefIdxActive}[i][j]$. La imagen a la que haga referencia cada entrada activa en $\text{RefPicList}[0][j]$ o $\text{RefPicList}[1][j]$ estará presente en el DPB y tendrá TemporalId menor o igual que el de la imagen actual. Opcionalmente, se especifica además la siguiente restricción: el índice de entrada de cualquier entrada inactiva en $\text{RefPicList}[0][j]$ o $\text{RefPicList}[1][j]$ no se utilizará como índice de referencia para la decodificación de la imagen actual. Opcionalmente, se especifica además la siguiente restricción: una entrada inactiva en $\text{RefPicList}[0][j]$ o $\text{RefPicList}[1][j]$ no se referirá a la misma imagen que cualquier otra entrada en $\text{RefPicList}[0][j]$ o $\text{RefPicList}[1][j]$. Una entrada STRP en $\text{RefPicList}[0][j]$ o $\text{RefPicList}[1][j]$ de una porción de una imagen y una entrada LTRP en $\text{RefPicList}[0][j]$ o $\text{RefPicList}[1][j]$ de la misma porción o de una porción diferente de la misma imagen no se referirán a la misma imagen. A la imagen actual en sí misma no hará referencia ninguna entrada en $\text{RefPicList}[0][j]$ o $\text{RefPicList}[1][j]$. No habrá ninguna entrada LTRP en $\text{RefPicList}[0][j]$ o $\text{RefPicList}[1][j]$ para la que la diferencia entre PicOrderCntVal de la imagen actual y PicOrderCntVal de la imagen a la que hace referencia la entrada sea superior o igual a 224. Sea setOfRefPics el conjunto de imágenes únicas a las que hacen referencia todas las entradas de $\text{RefPicList}[0][j]$ y todas las entradas de $\text{RefPicList}[1][j]$. El número de imágenes en setOfRefPics será menor o igual que $\text{sps_max_dec_pic_buffering_minus1}$, y setOfRefPics será el mismo para todas las porciones de una imagen.

Cada STRP se identifica por su valor PicOrderCntVal . Para cada LTRP, si hace referencia a la misma una entrada en $\text{RefPicList}[0][j]$ o $\text{RefPicList}[1][j]$ con $\text{delta_poc_msb_present_flag}[i][j]$ igual a 1, se identifica por su valor PicOrderCntVal , de lo contrario, se identifica por sus $\text{Log2}(\text{MaxPicOrderCntLsb})$ LSB de su valor PicOrderCntVal .

Alternativa 2 de señalización de MSB de delta POC para las entradas de imágenes de referencia a largo plazo.

En una realización alternativa de la divulgación, la divulgación descrita en la primera realización o la segunda realización se puede combinar con las realizaciones descritas anteriormente y denominadas "Señalización de MSB de delta POC para las entradas de imágenes de referencia a largo plazo" y "Alternativa 1 de señalización de MSB de delta POC para las entradas de imágenes de referencia a largo plazo", respectivamente. Los aspectos de las divulgaciones que se han de combinar son la señalización de $\text{additional_lt_poc_lsb}$ (es decir, de la primera realización o la segunda realización) e información del ciclo de MSB de POC (es decir, de la realización descrita anteriormente y denominada "Señalización de MSB de delta POC para las entradas de imágenes de referencia a largo plazo" o "Alternativa 1 de señalización de MSB de delta POC para las entradas de imágenes de referencia a largo plazo"). Un ejemplo de cómo se puede hacer la combinación, que combina la primera realización y la realización descrita anteriormente y denominada "Alternativa 1 de señalización de MSB de delta POC para las entradas de imágenes de referencia a largo plazo", se describe de la siguiente manera:

slice_header() {	Descriptor
slice_pic_parameter_set_id	ue(v)
slice_address	u(v)
slice_type	ue(v)
if (slice_type != I)	
log2_diff_ctu_max_bt_size	ue(v)
if(nal_unit_type != IRAP_NUT) {	
slice_pic_order_cnt_lsb	u(v)
for(i = 0; i < 2; i++) {	
ref_pic_list_sps_flag[i]	u(1)
if(ref_pic_list_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1)	
ref_pic_list_idx[i]	u(v)
} else	

ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i], long_term_ref_pics_flag)	
if(long_term_ref_pics_flag) {	
NumLtrpEntries[i] = ref_pic_list_sps_flag[i] ?	
num_ltrp_entries[i][ref_pic_list_idx[i]] :	
num_ltrp_entries[i][num_ref_pic_lists_in_sps[i]]	
for(j = 0; j < NnumLtrpEntries[i]; j++) {	
delta_poc_msb_present_flag[i][j]	u(1)
if(delta_poc_msb_present_flag[i][j])	
delta_poc_msb_cycle_lt[i][j]	ue(v)
}	
}	
if(slice_type == P slice_type == B) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag)	
for(i = 0; i < (slice_type == B ? 2 : 1); i++)	
num_ref_idx_active_minus1[i]	ue(v)
}	
}	
byte_alignment()	
}	

delta_poc_msb_present_flag[i][j] igual a 1 especifica que delta_pocmsb_cycle_lt[i][j] está presente. delta_pocmsb_present_flag[i][i] igual a 0 especifica que delta_poc_msb_cycle_lt[i][j] no está presente. Cuando NumLtrpEntries[i] sea mayor de 0 y para la j-ésima entrada LTRP en la estructura de sintaxis ref_pic_list_struct(i, rplIdx, 1) haya más de una imagen de referencia en el DPB en el momento en que se decodifica este encabezado de porción para las que el módulo MaxPicOrderLtCntLsb de PicOrderCntVal sea igual a poc_lsb_lt[i][rplIdx][jj], donde jj es el índice de entrada de la entrada en la estructura de sintaxis ref_pic_list_struct(i, rplIdx, 1) que es la j-ésima entrada LTRP en la estructura de sintaxis ref_pic_list_struct(i, rplIdx, 1), delta_pocmsb_present_flag[i][j] será igual a 1. Cuando no está presente, se deduce que el valor de delta_poc_msb_cycle_lt[i][j] es igual a 0.

delta_poc_msb_cycle_lt[i][j] se utiliza para determinar el valor de los bits más significativos del valor de recuento de orden de imágenes de la j-ésima entrada LTRP en la estructura de sintaxis ref_pic_list_struct(i, rplIdx, 1). Cuando delta_poc_msb_cycle_lt[i][j] no está presente, se deduce que es igual a 0.

Cambios en el proceso de decodificación para el recuento de orden de imágenes:

Las listas RefPicList[0] y RefPicList[1] de imágenes de referencia se construyen de la siguiente manera:

```

for( i = 0; i < 2; i++ ) {
    if( ref_pic_list_sps_flag[ i ] )
        RplIdx[ i ] = ref_pic_list_idx[ i ]
    else
        RplIdx[ i ] = num_ref_pic_lists_in_sps[ i ]
    msbCycleIdx = 0
    for( j = 0; pocBase = PicOrderCntVal; j < NumEntriesInList[ i ][ RplIdx[ i ] ]; j++ ) {
        if( !h_ref_pic_flag[ i ][ RplIdx[ i ] ][ j ] )
            RefPicPoolList[ i ][ j ] = pocBase - delta_poc_s[ i ][ RplIdx[ i ] ][ j ]
        if( hay una imagen picA de referencia en el DPB con PicOrderCntVal igual a RefPicPoolList[ i ][ j ] )
            RefPicList[ i ][ j ] = picA
        else
            RefPicList[ i ][ j ] = "ninguna imagen de referencia"
        pocBase = RefPicPoolList[ i ][ j ]
    } else {
        if( delta_poc_msb_cycle_it[ i ][ msbCycleIdx ] es igual a 0 &&
            hay una imagen picA de referencia en el DPB con PicOrderCntVal &
            ( MaxLtpPicOrderCntLsb - 1 ) igual a poc_lsb_it[ i ][ RplIdx[ i ] ][ j ] )
            RefPicList[ i ][ j ] = picA
        else if( delta_poc_msb_cycle_it[ i ][ msbCycleIdx ] es igual a 1 &&
            hay una imagen picA de referencia en el DPB con
            PicOrderCntVal igual a (
            MaxPicOrderCntLsb +
            delta_poc_msb_cycle_it[ i ][ msbCycleIdx ] + poc_lsb_it[ i ][ RplIdx[ i ] ][ j ] ) )
            RefPicList[ i ][ j ] = picA
        msbCycleIdx++
    } else
        RefPicList[ i ][ j ] = "ninguna imagen de referencia"
    }
}
}
}

```

Como alternativa, la semántica de `delta_poc_msb_cycle_it[ listIdx ][ i ]` puede expresarse como delta de delta, de manera que la construcción de listas de imágenes de referencia se puede actualizar de la siguiente manera: las listas `RefPicList[ 0 ]` y `RefPicList[ 1 ]` de imágenes de referencia se construyen de la siguiente manera:

```

for( i = 0; i < 2; i++) {
    if( ref_pic_list_sps_flag[ i ] )
        RplIdx[ i ] = ref_pic_list_idx[ i ]
    else
        RplIdx[ i ] = num_ref_pic_lists_in_sps[ i ]
    prevMsbCycle = 0
    msbCycleIdx = 0
    for( j = 0; pocBase = PicOrderCntVal; j < NumEntriesInList[ i ][ RplIdx[ i ] ]++; ) {
        if( ! ref_pic_flag[ i ][ RplIdx[ i ] ][ j ] ) {
            RefPicPocList[ i ][ j ] = pocBase + delta_poc_s[ i ][ RplIdx[ i ] ][ j ]
            if( hay una imagen picA de referencia en el DPB con PicOrderCntVal igual a RefPicPocList[ i ][ j ] )
                RefPicList[ i ][ j ] = picA
        }
        else
            RefPicList[ i ][ j ] = "ninguna imagen de referencia"
            pocBase = RefPicPocList[ i ][ j ]
        } else {
            if( delta_poc_msb_cycle_0[ i ][ msbCycleIdx ] es igual a 0 &&
                hay una imagen picA de referencia en el DPB con PicOrderCntVal &
                ( MaxLtPicOrderCntLsb - 1 ) igual a poc_lsb_0[ i ][ RplIdx[ i ] ][ j ] )
                RefPicList[ i ][ j ] = picA
            else if( delta_poc_msb_cycle_1[ i ][ msbCycleIdx ] es igual a 1 &&
                hay una imagen picA de referencia en el DPB con PicOrderCntVal igual a
                ( MaxPicOrderCntLsb * ( delta_poc_msb_cycle_1[ i ][ msbCycleIdx ] +
                    prevMsbCycle ) + poc_lsb_1[ i ][ RplIdx[ i ] ][ j ] ) )
                RefPicList[ i ][ j ] = picA
            prevMsbCycle += delta_poc_msb_cycle_1[ i ][ msbCycleIdx ]
            msbCycleIdx++
        } else
            RefPicList[ i ][ j ] = "ninguna imagen de referencia"
        }
    }
}

```

Es un requisito de conformidad de flujo de bits que se apliquen las siguientes restricciones: para cada i igual a 0 o 1, $\text{NumEntriesInList}[i][\text{RplIdx}[i]]$ no será menor que $\text{NumRefIdxActive}[i]$. La imagen a la que haga referencia cada entrada activa en $\text{RefPicList}[0]$ o $\text{RefPicList}[1]$ estará presente en el DPB y tendrá TemporalId menor o igual que el de la imagen actual. Opcionalmente, se especifica además la siguiente restricción: el índice de entrada de cualquier entrada inactiva en $\text{RefPicList}[0]$ o $\text{RefPicList}[1]$ no se utilizará como índice de referencia para la decodificación de la imagen actual. Opcionalmente, se especifica además la siguiente restricción: una entrada inactiva en $\text{RefPicList}[0]$ o $\text{RefPicList}[1]$ no se referirá a la misma imagen que cualquier otra entrada en $\text{RefPicList}[0]$ o $\text{RefPicList}[1]$. Una entrada STRP en $\text{RefPicList}[0]$ o $\text{RefPicList}[1]$ de una porción de una imagen y una entrada LTRP en $\text{RefPicList}[0]$ o $\text{RefPicList}[1]$ de la misma porción o de una porción diferente de la misma imagen no se referirán a la misma imagen. A la imagen actual en sí misma no hará referencia ninguna entrada en $\text{RefPicList}[0]$ o $\text{RefPicList}[1]$. No habrá ninguna entrada LTRP en $\text{RefPicList}[0]$ o $\text{RefPicList}[1]$ para la que la diferencia entre PicOrderCntVal de la imagen actual y PicOrderCntVal de la imagen a la que hace referencia la entrada sea superior o igual a 224. Sea setOfRefPics el conjunto de imágenes únicas a las que hacen referencia todas las entradas de $\text{RefPicList}[0]$ y todas las entradas de $\text{RefPicList}[1]$. El número de imágenes en setOfRefPics será menor o igual que $\text{sps_max_dec_pic_buffering_minus1}$, y setOfRefPics será el mismo para todas las porciones de una imagen.

Cada STRP se identifica por su valor PicOrderCntVal . Para cada LTRP, si hace referencia a la misma una entrada en $\text{RefPicList}[0]$ o $\text{RefPicList}[1]$ con $\text{delta_poc_msb_present_flag}[i][j]$ igual a 1, se identifica por su valor PicOrderCntVal , de lo contrario, se identifica por sus $\text{Log2}(\text{MaxLtPicOrderCntLsb})$ LSB de su valor PicOrderCntVal .

Señalización siempre de listas de imágenes de referencia en encabezados de porción con diferenciación entre imágenes de referencia a corto y largo plazo.

Esta sección describe otra realización alternativa de la divulgación. La descripción es relativa al último WD de VVC (es decir, sólo se describe el delta relativo al último WD de VVC en JVET-K1001-v1, mientras que los textos en el último WD de VVC que no se mencionan posteriormente se aplican como son). Esta realización alternativa se resume de la siguiente manera: las estructuras de lista de imágenes de referencia se señalizan sólo en encabezados de porción. Tanto las imágenes de referencia a corto plazo como las imágenes de referencia a largo plazo se identifican mediante sus LSB de POC, que están representados por números de bits que son diferentes del número de bits utilizado para representar los LSB de POC señalizados en encabezados de porción para la obtención de valores de POC. Además, los números de bits utilizados para representar los LSB de POC para identificar imágenes de referencia

a corto plazo e imágenes de referencia a largo plazo son diferentes.

Sintaxis de encabezado de unidad NAL.

nal_unit_header() {	Descriptor
forbidden_zero_bit	f(1)
nal_unit_type	u(5)
nuh_temporal_id_plus1	u(3)
nuh_reserved_zero_7bits	u(7)
}	

5

Sintaxis de RBSP de conjunto de parámetros de secuencia.

seq_parameter_set_rbsp() {	Descriptor
sps_seq_parameter_set_id	ue(v)
chroma_format_idc	ue(v)
if(chroma_format_idc == 3)	
separate_colour_plane_flag	u(1)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
bit_depth_luma_minus8	ue(v)
bit_depth_chroma_minus8	ue(v)
log2_max_pic_order_cnt_lsb_minus4	ue(v)
sps_max_dec_pic_buffering_minus1	ue(v)
qtbtt_dual_tree_intra_flag	ue(v)
log2_ctu_size_minus2	ue(v)
log2_min_qt_size_intra_slices_minus2	ue(v)
log2_min_qt_size_inter_slices_minus2	ue(v)
max_mtt_hierarchy_depth_inter_slices	ue(v)
max_mtt_hierarchy_depth_intra_slices	ue(v)
additional_st_poc_lsb	ue(v)
long_term_ref_pics_flag	u(1)
if(long_term_ref_pics_flag)	
additional_lt_poc_lsb	ue(v)
rbsp_trailing_bits()	
}	

Sintaxis de RBSP de conjunto de parámetros de imagen.

10

pic_parameter_set_rbsp() {	Descriptor
pps_pic_parameter_set_id	ue(v)
pps_seq_parameter_set_id	ue(v)
for(i = 0; i < 2; i++)	
num_ref_idx_default_active_minus1[i]	ue(v)
rbsp_trailing_bits()	
}	

Sintaxis de encabezado de porción.

slice_header() {	Descriptor
slice_pic_parameter_set_id	ue(v)
slice_address	u(v)
slice_type	ue(v)
if(slice_type != I)	
log2_diff_ctu_max_bt_size	ue(v)
if(nal_unit_type != IRAP_NUT) {	
slice_pic_order_cnt_lsb	u(v)
for(i = 0; i < 2; i++)	
ref_pic_list_struct(i, long_term_ref_pics_flag)	
if(slice_type == P slice_type == B) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag)	
for(i = 0; i < (slice_type == B ? 2 : 1); i++)	

num_ref_idx_active_minus1[i]	ue(v)
}	
}	
byte_alignment()	
}	

Sintaxis de estructura de lista de imágenes de referencia.

ref_pic_list_struct(listIdx, ltrpFlag) {	Descriptor
num_strp_entries[listIdx]	ue(v)
if(ltrpFlag)	
num_ltrp_entries[listIdx]	ue(v)
for(i = 0; i < NumEntriesInList[listIdx]; i++) {	
if(num_ltrp_entries[listIdx] > 0)	
lt_ref_pic_flag[listIdx][i]	
if(!lt_ref_pic_flag[listIdx][i])	
poc_lsb_st[listIdx][i]	u(v)
else	
poc_lsb_lt[listIdx][i]	u(v)
}	
}	

5 Semántica de encabezado de unidad NAL.

Un forbidden_zero_bit será igual a 0. nal_unit_type especifica el tipo de estructura de datos de RBSP contenida en la unidad NAL.

10 Tabla 7-1 - Códigos de tipo de unidad NAL y clases de tipo de unidad NAL

nal_unit_type	Nombre de nal_unit_type	Contenido de unidad NAL y estructura de sintaxis de RBSP	Clase de tipo de unidad NAL
0	NON_IRAP_NUT	Segmento de porción codificada de una imagen no-IRAP slice_layer_rbsp()	VCL
1	IRAP_NUT	Porción codificada de una imagen IRAP slice_layer_rbsp()	VCL
2-15	RSV_VCL_NUT	Unidades NAL de VCL reservadas	VCL
16	SPS_NUT	Conjunto de parámetros de secuencia seq_parameter_set_rbsp()	no-VCL
17	PPS_NUT	Conjunto de parámetros de imagen pic_parameter_set_rbsp()	no-VCL
18	EOS_NUT	Fin de secuencia end_of_seq_rbsp()	no-VCL
19	EOB_NUT	Fin de flujo de bits end_of_bitstream_rbsp()	no-VCL
20, 21	PREFIX_SEI_NUT SUFFIX_SEI_NUT	Información de mejora suplementaria sei_rbsp()	no-VCL
22-26	RSV_NVCL	Reservados	no-VCL
27-31	UNSPEC	Sin especificar	no-VCL

Un nuh_temporal_id_plus1 minus 1 especifica un identificador temporal para la unidad NAL. El valor de nuh_temporal_id_plus1 no será igual a 0. La variable TemporalId se especifica como sigue: TemporalId = nuh_temporal_id_plus1 - 1.

- 15 Cuando nal_unit_type es igual a IRAP_NUT, la porción codificada pertenece a una imagen IRAP y TemporalId será igual a 0. El valor de TemporalId será el mismo para todas las unidades NAL de VCL de una unidad de acceso. El valor de TemporalId de una imagen codificada o de una unidad de acceso es el valor de TemporalId de las unidades NAL de VCL de la imagen codificada o de la unidad de acceso. El valor de TemporalId para las unidades NAL no-VCL se limita de la siguiente manera: si nal_unit_type es igual a SPS_NUT, TemporalId será igual a 0 y TemporalId de la unidad de acceso que contiene la unidad NAL será igual a 0. De lo contrario, si nal_unit_type es igual a EOS_NUT o EOB_NUT, TemporalId será igual a 0. De lo contrario, TemporalId será mayor o igual que TemporalId de la unidad de acceso que contiene la unidad NAL. Cuando la unidad NAL es una unidad NAL no-VCL, el valor de TemporalId es igual al valor mínimo de los valores de TemporalId de todas las unidades de acceso a las que se aplica la unidad NAL no-VCL. Cuando nal_unit_type es igual a PPS_NUT, TemporalId es mayor o igual que TemporalId de la unidad de acceso continente, ya que todos los conjuntos de parámetros de imagen (PPS) se incluyen en el comienzo de un flujo de bits, en donde la primera imagen codificada tiene TemporalId igual a 0. Cuando nal_unit_type es igual a
- 20
- 25

PREFIX_SEI_NUT o SUFFIX_SEI_NUT, TemporalId es mayor o igual que TemporalId de la unidad de acceso continente, ya que una unidad NAL de información de mejora suplementaria (SEI, por sus siglas en inglés) contiene información que se aplica a un subconjunto de flujo de bits que incluye unidades de acceso para las que los valores de TemporalId son mayores que TemporalId de la unidad de acceso que contiene la unidad NAL de SEI.

5 nuh_reserved_zero_7bits será igual a '0000000'. Otros valores de nuh_reserved_zero_7bits serán especificados en el futuro por ITU-T | ISO/IEC. Los decodificadores deberán ignorar (es decir, eliminar del flujo de bits y descartar) las unidades NAL con valores de nuh_reserved_zero_7bits no iguales a '0000000'.

Semántica de RBSP de conjunto de parámetros de secuencia.

10 Un log2_max_pic_order_cnt_lsb_minus4 especifica el valor de la variable MaxPicOrderCntLsb que se utiliza en el proceso de decodificación para el recuento de orden de imágenes de la siguiente manera:

$$\text{MaxPicOrderCntLsb} = 2^{(\log_2 \text{max_pic_order_cnt_lsb_minus4} + 4)}$$

15 El valor de log2_max_pic_order_cnt_lsb_minus4 estará en el intervalo de 0 a 12, inclusive. sps_max_dec_pic_buffering_minus1 plus 1 especifica el tamaño máximo requerido del búfer de imágenes decodificadas para la CVS en unidades de búferes de almacenamiento de imágenes. El valor de sps_max_dec_pic_buffering_minus1 estará en el intervalo de 0 a MaxDpbSize - 1, inclusive, donde MaxDpbSize es como se especifica en otro lugar.

20 additional_st_poc_lsb especifica el valor de la variable MaxStPicOrderCntLsb que se utiliza en el proceso de decodificación para las listas de imágenes de referencia de la siguiente manera:

$$\text{MaxStPicOrderCntLsb} = 2^{(\log_2 \text{max_pic_order_cnt_lsb_minus4} + 4 + \text{additional_st_poc_lsb})}$$

25 El valor de additional_st_poc_lsb estará en el intervalo de 0 a 32 - log2_max_pic_order_cnt_lsb_minus4 - 4, inclusive. long_term_ref_pics_flag igual a 0 especifica que no se usa LTRP para la interpredicción de ninguna imagen codificada en la CVS. long_term_ref_pics_flag igual a 1 especifica que se usan LTRP para la interpredicción de una o más imágenes codificadas en la CVS. additional_lt_poc_lsb especifica el valor de la variable MaxLtPicOrderCntLsb que se usa en el proceso de decodificación para las listas de imágenes de referencia de la siguiente manera:

$$\text{MaxLtPicOrderCntLsb} = 2^{(\log_2 \text{max_pic_order_cnt_lsb_minus4} + 4 + \text{additional_st_poc_lsb} + \text{additional_lt_poc_lsb})}$$

35 El valor de additional_lt_poc_lsb estará en el intervalo de 0 a 32 - log2_max_pic_order_cnt_lsb_minus4 - 4 - additional_st_poc_lsb, inclusive. Cuando no está presente, se deduce que el valor de additional_lt_poc_lsb es igual a 0.

Semántica de RBSP de conjunto de parámetros de imagen.

40 Un num_ref_idx_default_active_minus1[i] plus 1, cuando i es igual a 0, especifica el valor deducido de la variable NumRefIdxActive[0] para las porciones P o B con num_ref_idx_active_override_flag igual a 0 y, cuando i es igual a 1, especifica el valor deducido de NumRefIdxActive[1] para las porciones B con num_ref_idx_active_override_flag igual a 0. El valor de num_ref_idx_default_active_minus1[i] estará en el intervalo de 0 a 14, inclusive.

Semántica de encabezado de porción.

45 Cuando estén presentes, el valor de cada uno de los elementos de sintaxis de encabezado de porción slice_pic_parameter_set_id y slice_pic_order_cnt_lsb será el mismo en todos los encabezados de porción de una imagen codificada. slice_type especifica el tipo de codificación de la porción de acuerdo con la Tabla 7-3.

50 Tabla 7-3 - Asociación de nombres a slice_type

slice_type	Nombre de slice_type
0	B (porción B)
1	P (porción P)
2	I (porción I)

Cuando nal_unit_type sea igual a IRAP_NUT, es decir, la imagen es una imagen IRAP, slice_type será igual a 2.

55 Un slice_pic_order_cnt_lsb especifica el módulo de recuento de orden de imágenes MaxPicOrderCntLsb para la imagen actual. La longitud del elemento de sintaxis slice_pic_order_cnt_lsb es log2_max_pic_order_cnt_lsb_minus4 + 4 bits. El valor de slice_pic_order_cnt_lsb estará en el intervalo de 0 a MaxPicOrderCntLsb - 1, inclusive. Cuando slice_pic_order_cnt_lsb no está presente, se deduce que slice_pic_order_cnt_lsb es igual a 0.

60 num_ref_idx_active_override_flag igual a 1 especifica que el elemento de sintaxis num_ref_idx_active_minus1[0] está presente para las porciones P y B y que el elemento de sintaxis num_ref_idx_active_minus1[1] está presente para las porciones B. num_ref_idx_active_override_flag igual a 0 especifica que los elementos de sintaxis

num_ref_idx_active_minus1[0] y num_ref_idx_active_minus1[1] no están presentes. num_ref_idx_active_minus1[i], cuando está presente, especifica el valor de la variable NumRefIdxActive[i] de la siguiente manera:

$$\text{NumRefIdxActive}[i] = \text{num_ref_idx_active_minus1}[i] + 1$$

El valor de num_ref_idx_active_minus1[i] estará en el intervalo de 0 a 14, inclusive. El valor de NumRefIdxActive[i] - 1 especifica el índice de referencia máximo para la lista i de imágenes de referencia que se utiliza para decodificar la porción. Cuando el valor de NumRefIdxActive[i] es igual a 0, no se utiliza ningún índice de referencia para la lista i de imágenes de referencia para decodificar la porción. Para i igual a 0 o 1, cuando la porción actual es una porción B y num_ref_idx_active_override_flag es igual a 0, se deduce que NumRefIdxActive[i] es igual a num_ref_idx_default_active_minus1[i] + 1. Cuando la porción actual es una porción P y num_ref_idx_active_override_flag es igual a 0, se deduce que NumRefIdxActive[0] es igual a num_ref_idx_default_active_minus1[0] + 1. Cuando la porción actual es una porción P, se deduce que NumRefIdxActive[1] es igual a 0. Cuando la porción actual es una porción I, se deduce que tanto NumRefIdxActive[0] como NumRefIdxActive[1] son iguales a 0. Como alternativa, para i igual a 0 o 1, se aplica lo siguiente después de lo anterior: sea rplIdx1 establecido como igual a ref_pic_list_sps_flag[i] ? ref_pic_list_idx[i] : num_ref_pic_lists_in_sps[i], y sea numRpEntries[i] igual a num_strp_entries[i][rplIdx1] + num_ltrp_entries[i][rplIdx1]. Cuando NumRefIdxActive[i] es mayor que numRpEntries[i], el valor de NumRefIdxActive[i] se establece igual a numRpEntries[i].

Semántica de estructura de lista de imágenes de referencia.

La estructura de sintaxis ref_pic_list_struct(listIdx, ltrpFlag) está presente en un encabezado de porción. Cuando está presente en un encabezado de porción, la estructura de sintaxis ref_pic_list_struct(listIdx, ltrpFlag) especifica la lista listIdx de imágenes de referencia de la imagen actual (la imagen que contiene la porción). num_strp_entries[listIdx] especifica el número de entradas STRP en la estructura de sintaxis ref_pic_list_struct(listIdx, ltrpFlag). num_ltrp_entries[listIdx] especifica el número de entradas LTRP en la estructura de sintaxis ref_pic_list_struct(listIdx, ltrpFlag). Cuando no está presente, se deduce que el valor de num_ltrp_entries[listIdx] es igual a 0. La variable NumEntriesInList[listIdx] se obtiene de la siguiente manera:

$$\text{NumEntriesInList}[\text{listIdx}] = \text{num_strp_entries}[\text{listIdx}] + \text{num_ltrp_entries}[\text{listIdx}]$$

El valor de NumEntriesInList[listIdx] estará en el intervalo de 0 a sps_max_dec_pic_buffering_minus1, inclusive. It_ref_pic_flag[listIdx][i] igual a 1 especifica que la i-ésima entrada en la estructura de sintaxis ref_pic_list_struct(listIdx, ltrpFlag) es una entrada LTRP. It_ref_pic_flag[listIdx][i] igual a 0 especifica que la i-ésima entrada en la estructura de sintaxis ref_pic_list_struct(listIdx, ltrpFlag) es una entrada STRP. Cuando no está presente, se deduce que el valor de It_ref_pic_flag[listIdx][i] es igual a 0. Es un requisito de conformidad de flujo de bits que la suma de It_ref_pic_flag[listIdx][i] para todos los valores de i en el intervalo de 0 a NumEntriesInList[listIdx] - 1, inclusive, sea igual a num_ltrp_entries[listIdx]. poc_lsb_st[listIdx][i], cuando It_ref_pic_flag[listIdx][i] es igual a 0, especifica el valor del módulo de recuento de orden de imágenes MaxStPicOrderCntLsb de la imagen a la que hace referencia la i-ésima entrada en la estructura de sintaxis ref_pic_list_struct(listIdx, ltrpFlag). La longitud del elemento de sintaxis poc_lsb_st[listIdx][i] es Log2(MaxStPicOrderCntLsb) bits. poc_lsb_lt[listIdx][i], cuando It_ref_pic_flag[listIdx][i] es igual a 1, especifica el valor del módulo de recuento de orden de imágenes MaxLtPicOrderCntLsb de la imagen a la que hace referencia la i-ésima entrada en la estructura de sintaxis ref_pic_list_struct(listIdx, ltrpFlag). La longitud del elemento de sintaxis poc_lsb_lt[listIdx][i] es Log2(MaxLtPicOrderCntLsb) bits.

Se expone el proceso de decodificación.

Proceso general de decodificación.

El proceso de decodificación funciona de la siguiente manera para la imagen actual CurrPic: la decodificación de las unidades NAL se especifica posteriormente. Los procesos expuestos posteriormente especifican los siguientes procesos de decodificación utilizando elementos de sintaxis en la capa de encabezado de porción y superiores: se obtienen variables y funciones en relación con el recuento de orden de imágenes. Es necesario invocar esto sólo para la primera porción de una imagen. Al comienzo del proceso de decodificación para cada porción de una imagen no-IRAP, se invoca el proceso de decodificación para la construcción de listas de imágenes de referencia para la obtención de la lista 0 de imágenes de referencia (RefPicList[0]) y la lista 1 de imágenes de referencia (RefPicList[1]). Se invoca el proceso de decodificación para el marcado de imágenes de referencia, en donde las imágenes de referencia se marcan como "no utilizadas para referencia" o "utilizadas para referencia a largo plazo". Es necesario invocar esto sólo para la primera porción de una imagen. Se invocan los procesos de decodificación para codificar unidades de árbol, cambiar la escala, transformar, filtrar en bucle, etc. Después de que se hayan decodificado todas las porciones de la imagen actual, la imagen decodificada actual se marca como "utilizada para referencia a corto plazo".

Proceso de decodificación de unidades NAL.

Las entradas a este proceso son unidades NAL de la imagen actual y sus unidades NAL no-VCL asociadas. Las salidas de este proceso son las estructuras de sintaxis de RBSP analizadas encapsuladas dentro de las unidades NAL. El proceso de decodificación para cada unidad NAL extrae la estructura de sintaxis de RBSP de la unidad NAL y, a continuación, analiza la estructura de sintaxis de RBSP.

Proceso de decodificación de porciones.

Proceso de decodificación para el recuento de orden de imágenes.

La salida de este proceso es $PicOrderCntVal$, el recuento de orden de imágenes de la imagen actual. Los recuentos de orden de imágenes se utilizan para identificar imágenes, para obtener parámetros de movimiento en modo de fusión y la predicción de vectores de movimiento, y para la verificación de conformidad del decodificador. Cada imagen codificada está asociada con una variable de recuento de orden de imágenes, denominada $PicOrderCntVal$. Cuando la imagen actual no es una imagen IRAP, las variables $prevPicOrderCntLsb$ y $prevPicOrderCntMsb$ se obtienen de la siguiente manera: sea $prevTid0Pic$ la imagen anterior en orden de decodificación que tenga $TemporalId$ igual a 0. La variable $prevPicOrderCntLsb$ se establece como igual a $slice_pic_order_cnt_lsb$ de $prevTid0Pic$. La variable $prevPicOrderCntMsb$ se establece como igual a $PicOrderCntMsb$ de $prevTid0Pic$. La variable $PicOrderCntMsb$ de la imagen actual se obtiene de la siguiente manera: si la imagen actual es una imagen IRAP, $PicOrderCntMsb$ se establece como igual a 0. De lo contrario, $PicOrderCntMsb$ se obtiene de la siguiente manera:

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

$PicOrderCntVal$ se obtiene de la siguiente manera:

```
PicOrderCntVal = PicOrderCntMsb + slice_pic_order_cnt_lsb
```

Todas las imágenes IRAP tendrán $PicOrderCntVal$ igual a 0, ya que se deduce que $slice_pic_order_cnt_lsb$ es 0 para las imágenes IRAP y $prevPicOrderCntLsb$ y $prevPicOrderCntMsb$ se establecen ambos como iguales a 0. El valor de $PicOrderCntVal$ estará en el intervalo de -231 a $231 - 1$, inclusive. En una CVS, los valores de $PicOrderCntVal$ para dos imágenes codificadas cualesquiera no serán los mismos. En cualquier momento durante el proceso de decodificación, los valores de $PicOrderCntVal \& (MaxStPicOrderCntLsb - 1)$ para dos imágenes de referencia a corto plazo cualesquiera en el DPB no serán los mismos. En cualquier momento durante el proceso de decodificación, los valores de $PicOrderCntVal \& (MaxLtPicOrderCntLsb - 1)$ para dos imágenes de referencia cualesquiera en el DPB no serán los mismos.

La función $PicOrderCnt(picX)$ se especifica de la siguiente manera:

```
PicOrderCnt( picX ) = PicOrderCntVal de la imagen picX
```

La función $DiffPicOrderCnt(picA, picB)$ se especifica de la siguiente manera:

```
DiffPicOrderCnt( picA, picB ) = PicOrderCnt( picA ) - PicOrderCnt( picB )
```

El flujo de bits no contendrá datos que den como resultado valores de $DiffPicOrderCnt(picA, picB)$ utilizados en el proceso de decodificación que no estén en el intervalo de -2^{15} a $2^{15} - 1$, inclusive. Sea X la imagen actual y sean Y y Z otras dos imágenes en la misma CVS, Y y Z se considera que están en la misma dirección de orden de salida de X cuando $DiffPicOrderCnt(X, Y)$ y $DiffPicOrderCnt(X, Z)$ son ambos positivos o ambos negativos.

Proceso de decodificación para la construcción de listas de imágenes de referencia.

Este proceso se invoca al principio del proceso de decodificación para cada porción de una imagen no-IRAP. Las imágenes de referencia se direccionan a través de índices de referencia. Un índice de referencia es un índice en una lista de imágenes de referencia. Al decodificar una porción I , no se utiliza ninguna lista de imágenes de referencia en la decodificación de los datos de la porción. Al decodificar una porción P , sólo se utiliza la lista 0 de imágenes de referencia (es decir, $RefPicList[0]$) en la decodificación de los datos de la porción. Al decodificar una porción B , tanto la lista 0 de imágenes de referencia como la lista 1 de imágenes de referencia (es decir, $RefPicList[1]$) se utilizan en la decodificación de los datos de la porción. Al comienzo del proceso de decodificación para cada porción de una imagen no-IRAP, se obtienen las listas $RefPicList[0]$ y $RefPicList[1]$ de imágenes de referencia. Las listas de

imágenes de referencia se utilizan en el marcado de imágenes de referencia o en la decodificación de los datos de la porción. Para una porción I de una imagen no-IRAP que no sea la primera porción de la imagen, RefPicList[0] y RefPicList[1] se obtienen con el propósito de verificar la conformidad del flujo de bits, pero su obtención no es necesaria para decodificar la o las imágenes actuales que siguen a la imagen actual en orden de decodificación. Para una porción P que no sea la primera porción de una imagen, RefPicList[1] se obtiene con el propósito de verificar la conformidad del flujo de bits, pero su obtención no es necesaria para decodificar la o las imágenes actuales que siguen a la imagen actual en orden de decodificación.

Las listas RefPicList[0] y RefPicList[1] de imágenes de referencia se construyen de la siguiente manera:

```

for( i = 0; i < 2; i++ ) {
    for( j = 0; j < NumEntriesInList[ i ]; j++ ) {
        if( !lt_ref_pic_flag[ i ][ j ] ) {
            if( hay una picA de referencia en el DPB con PicOrderCntVal & ( MaxLtpicOrderCntLsb - 1 )
                igual a poc_isb_#( i ][ j ] ) ) {
                RefPicList[ i ][ j ] = picA;
            }
            else {
                RefPicList[ i ][ j ] = "ninguna imagen de referencia";
            }
        }
    }
}
for( i = 0; i < 2; i++ ) {
    for( j = 0; j < NumEntriesInList[ i ]; j++ ) {
        if( !lt_ref_pic_flag[ i ][ j ] ) {
            if( hay una imagen picA de referencia a como plazo en el DPB
                con PicOrderCntVal & ( MaxStpicOrderCntLsb - 1 ) igual a poc_isb_#( i ][ j ] ) ) {
                RefPicList[ i ][ j ] = picA;
            }
            else {
                RefPicList[ i ][ j ] = "ninguna imagen de referencia";
            }
        }
    }
}

```

Para cada i igual a 0 o 1, se aplica lo siguiente:

Las primeras entradas NumRefIdxActive[i] en RefPicList[i] se denominan entradas activas en RefPicList[i], y las otras entradas en RefPicList[i] se denominan entradas inactivas en RefPicList[i]. Cada entrada en RefPicList[i][j] para j en el intervalo de 0 a NumEntriesInList[i] - 1, inclusive, se conoce como una entrada STRP si lt_ref_pic_flag[i][j] es igual a 0, y como una entrada LTRP de lo contrario. Es posible que a una imagen en particular hagan referencia tanto una entrada en RefPicList[0] como una entrada en RefPicList[1]. También es posible que a una imagen en particular hagan referencia más de una entrada en RefPicList[0] o más de una entrada en RefPicList[1]. Las entradas activas en RefPicList[0] y las entradas activas en RefPicList[1] se refieren colectivamente a todas las imágenes de referencia que se utilizan para la interpredicción de la imagen actual y una o más imágenes que siguen a la imagen actual en orden de decodificación. Las entradas inactivas en RefPicList[0] y las entradas inactivas en RefPicList[1] se refieren colectivamente a todas las imágenes de referencia que no se utilizan para la interpredicción de la imagen actual, pero se utilizan para la interpredicción de una o más imágenes que siguen a la imagen actual en orden de decodificación. Hay una o más entradas en RefPicList[0] o RefPicList[1] que son iguales a "ninguna imagen de referencia" porque las imágenes correspondientes no están presentes en el DPB. Cada entrada inactiva en RefPicList[0] o RefPicList[0] que sea igual a "ninguna imagen de referencia" debe ignorarse. Se debe deducir una pérdida de imagen no intencionada para cada entrada activa en RefPicList[0] o RefPicList[1] que sea igual a "ninguna imagen de referencia".

Es un requisito de conformidad de flujo de bits que se apliquen las siguientes restricciones: para cada i igual a 0 o 1, NumEntriesInList[i] no será menor que NumRefIdxActive[i]. La imagen a la que haga referencia cada entrada activa en RefPicList[0] o RefPicList[1] estará presente en el DPB y tendrá TemporalId menor o igual que el de la imagen actual. Opcionalmente, se especifica además la siguiente restricción: el índice de entrada de cualquier entrada inactiva en RefPicList[0] o RefPicList[1] no se utilizará como índice de referencia para la decodificación de la imagen actual. Opcionalmente, se especifica además la siguiente restricción: una entrada inactiva en RefPicList[0] o RefPicList[1] no se referirá a la misma imagen que cualquier otra entrada en RefPicList[0] o RefPicList[1]. Una entrada STRP en RefPicList[0] o RefPicList[1] de una porción de una imagen y una entrada LTRP en RefPicList[0] o RefPicList[1] de la misma porción o de una porción diferente de la misma imagen no se referirán a la misma imagen. A la imagen actual en sí misma no hará referencia ninguna entrada en RefPicList[0] o RefPicList[1]. No habrá ninguna entrada LTRP en RefPicList[0] o RefPicList[1] para la que la diferencia entre PicOrderCntVal de la imagen actual y PicOrderCntVal de la imagen a la que hace referencia la entrada sea superior o igual a 224. Sea setOfRefPics el conjunto de imágenes únicas a las que hacen referencia todas las entradas de RefPicList[0] y todas las entradas de

RefPicList[1]. El número de imágenes en setOfRefPics será menor o igual que sps_max_dec_pic_buffering_minus1, y setOfRefPics será el mismo para todas las porciones de una imagen.

Proceso de decodificación para el marcado de imágenes de referencia.

- 5 Este proceso se invoca una vez por imagen, después de decodificar un encabezado de porción y el proceso de decodificación para la construcción de listas de imágenes de referencia para la porción, pero antes de la decodificación de los datos de la porción. Este proceso da como resultado que una o más imágenes de referencia en el DPB se marquen como "no utilizadas para referencia" o "utilizadas para referencia a largo plazo". Una imagen decodificada en el DPB puede marcarse como "no utilizada para referencia", "utilizada para referencia a corto plazo" o "utilizada para referencia a largo plazo", pero sólo una de estas tres en cualquier momento dado durante la operación del proceso de decodificación. Asignar una de estas marcas a una imagen elimina implícitamente otra de estas marcas cuando corresponda. Cuando se hace referencia a una imagen como marcada como "utilizada para referencia", esto se refiere colectivamente a que la imagen está marcada como "utilizada para referencia a corto plazo" o "utilizada para referencia a largo plazo" (pero no ambas). Cuando la imagen actual es una imagen IRAP, todas las imágenes de referencia actualmente presentes en el DPB (si las hay) se marcan como "no utilizadas para referencia". Las STRP se identifican mediante los $\text{Log}_2(\text{MaxStPicOrderCntLsb})$ LSB de sus valores PicOrderCntVal. Las LTRP se identifican mediante los $\text{Log}_2(\text{MaxLtPicOrderCntLsb})$ LSB de sus valores PicOrderCntVal.
- 10 Se aplica lo siguiente: para cada entrada LTRP en RefPicList[0] o RefPicList[1], cuando la imagen a la que se hace referencia es una STRP, la imagen se marca como "utilizada para referencia a largo plazo". Cada imagen de referencia en el DPB a la que no haga referencia ninguna entrada de RefPicList[0] o RefPicList[1] está marcada como "no utilizada para referencia".
- 15 Señalización siempre de listas de imágenes de referencia en encabezados de porción sin diferenciación entre imágenes de referencia a corto y largo plazo.
- 20 Esta sección describe otra realización alternativa de la divulgación. La descripción es relativa al último WD de VVC (es decir, sólo se describe el delta relativo al último WD de VVC en JVET-K1001-v1, mientras que los textos en el último WD de VVC que no se mencionan posteriormente se aplican como son). Esta realización alternativa se resume de la siguiente manera: las estructuras de lista de imágenes de referencia se señalizan sólo en encabezados de porción. No se hace distinción entre imágenes de referencia a corto y largo plazo. Todas las imágenes de referencia se denominan simplemente imágenes de referencia. Las imágenes de referencia se identifican por sus LSB de POC, que están representados por un número de bits que es diferente del número de bits utilizado para representar los LSB de POC señalizados en encabezados de porción para la obtención de valores de POC.
- 25 Abreviaturas. Se aplica el texto de la cláusula 4 del WD de VVC.
- 30 Sintaxis de encabezado de unidad NAL.
- 35

nal_unit_header() {	Descriptor
forbidden_zero_bit	f(1)
nal_unit_type	u(5)
nuh_temporal_id_plus1	u(3)
nuh_reserved_zero_7bits	u(7)
}	

Sintaxis de RBSP de conjunto de parámetros de secuencia.

seq_parameter_set_rbsp() {	Descriptor
sps_seq_parameter_set_id	ue(v)
chroma_format_idc	ue(v)
if(chroma_format_idc == 3)	
separate_colour_plane_flag	u(1)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
bit_depth_luma_minus8	ue(v)
bit_depth_chroma_minus8	ue(v)
log2_max_pic_order_cnt_lsb_minus4	ue(v)
sps_max_dec_pic_buffering_minus1	ue(v)
qtbtt_dual_tree_intra_flag	ue(v)
log2_ctu_size_minus2	ue(v)
log2_min_qt_size_intra_slices_minus2	ue(v)
log2_min_qt_size_inter_slices_minus2	ue(v)
max_mtt_hierarchy_depth_inter_slices	ue(v)

max_mtt_hierarchy_depth_intra_slices	ue(v)
additional_ref_poc_lsb	ue(v)
rbsp_trailing_bits()	
}	

Sintaxis de RBSP de conjunto de parámetros de imagen.

pic_parameter_set_rbsp() {	Descriptor
pps_pic_parameter_set_id	ue(v)
pps_seq_parameter_set_id	ue(v)
for(i = 0; i < 2; i++)	
num_ref_idx_default_active_minus1[i]	ue(v)
rbsp_trailing_bits()	
}	

5 Sintaxis de encabezado de porción.

slice_header() {	Descriptor
slice_pic_parameter_set_id	ue(v)
slice_address	u(v)
slice_type	ue(v)
if (slice_type != I)	
log2_diff_ctu_max_bt_size	ue(v)
if(nal_unit_type != IRAP_NUT) {	
slice_pic_order_cnt_lsb	u(v)
for(i = 0; i < 2; i++)	
ref_pic_list_struct(i)	
if(slice_type == P slice_type == B) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag)	
for(i = 0; i < (slice_type == B ? 2 : 1); i++)	
num_ref_idx_active_minus1[i]	ue(v)
}	
}	
byte_alignment()	
}	

Sintaxis de estructura de lista de imágenes de referencia.

ref_pic_list_struct(listIdx) {	Descriptor
num_ref_entries[listIdx]	ue(v)
for(i = 0; i < NumEntriesInList[listIdx]; i++)	
poc_ref_lsb[listIdx][i]	u(v)
}	

10

Semántica de encabezado de unidad NAL.

Un forbidden_zero_bit será igual a 0. nal_unit_type especifica el tipo de estructura de datos de RBSP contenida en la unidad NAL.

15

Tabla 7-1 - Códigos de tipo de unidad NAL y clases de tipo de unidad NAL

nal_unit_type	Nombre de nal_unit_type	Contenido de unidad NAL y estructura de sintaxis de RBSP	Clase de tipo de unidad NAL
0	NON_IRAP_NUT	Segmento de porción codificada de una imagen no-IRAP slice_layer_rbsp()	VCL
1	IRAP_NUT	Porción codificada de una imagen IRAP slice_layer_rbsp()	VCL
2-15	RSV_VCL_NUT	Unidades NAL de VCL reservadas	VCL
16	SPS_NUT	Conjunto de parámetros de secuencia seq_parameter_set_rbsp()	no-VCL
17	PPS_NUT	Conjunto de parámetros de imagen pic_parameter_set_rbsp()	no-VCL

18	EOS_NUT	Fin de secuencia end_of_seq_rbsp()	no-VCL
19	EOB_NUT	Fin de flujo de bits end_of_bitstream_rbsp()	no-VCL
20, 21	PREFIX_SEI_NUT SUFFIX_SEI_NUT	Información de mejora suplementaria sei_rbsp()	no-VCL
22-26	RSV_NVCL	Reservados	no-VCL
27-31	UNSPEC	Sin especificar	no-VCL

Un nuh_temporal_id_plus1 minus 1 especifica un identificador temporal para la unidad NAL. El valor de nuh_temporal_id_plus1 no será igual a 0. La variable TemporalId se especifica de la siguiente manera:

5 TemporalId = nuh_temporal_id_plus1 - 1

Cuando nal_unit_type es igual a IRAP_NUT, la porción codificada pertenece a una imagen IRAP, TemporalId será igual a 0. El valor de TemporalId será el mismo para todas las unidades NAL de VCL de una unidad de acceso. El valor de TemporalId de una imagen codificada o de una unidad de acceso es el valor de TemporalId de las unidades NAL de VCL de la imagen codificada o de la unidad de acceso. El valor de TemporalId para las unidades NAL no-VCL se limita de la siguiente manera:

Si nal_unit_type es igual a SPS_NUT, TemporalId será igual a 0 y TemporalId de la unidad de acceso que contiene la unidad NAL será igual a 0. De lo contrario, si nal_unit_type es igual a EOS_NUT o EOB_NUT, TemporalId será igual a 0. De lo contrario, TemporalId será mayor o igual que TemporalId de la unidad de acceso que contiene la unidad NAL. Cuando la unidad NAL es una unidad NAL no-VCL, el valor de TemporalId es igual al valor mínimo de los valores de TemporalId de todas las unidades de acceso a las que se aplica la unidad NAL no-VCL. Cuando nal_unit_type es igual a PPS_NUT, TemporalId es mayor o igual que TemporalId de la unidad de acceso continente, ya que todos los conjuntos de parámetros de imagen (PPS) se incluyen en el comienzo de un flujo de bits, en donde la primera imagen codificada tiene TemporalId igual a 0. Cuando nal_unit_type es igual a PREFIX_SEI_NUT o SUFFIX_SEI_NUT, TemporalId es mayor o igual que TemporalId de la unidad de acceso continente, ya que una unidad NAL de SEI contiene información que se aplica a un subconjunto de flujo de bits que incluye unidades de acceso para las que los valores de TemporalId son mayores que TemporalId de la unidad de acceso que contiene la unidad NAL de SEI. nuh_reserved_zero_7bits será igual a '0000000'. Otros valores de nuh_reserved_zero_7bits serán especificados en el futuro por ITU-T | ISO/IEC. Los decodificadores deberán ignorar (es decir, eliminar del flujo de bits y descartar) las unidades NAL con valores de nuh_reserved_zero_7bits no iguales a '0000000'.

Semántica de RBSP de conjunto de parámetros de secuencia.

30 log2_max_pic_order_cnt_lsb_minus4 especifica el valor de la variable MaxPicOrderCntLsb que se utiliza en el proceso de decodificación para el recuento de orden de imágenes de la siguiente manera:

$$\text{MaxPicOrderCntLsb} = 2^{(\log_2 \text{max_pic_order_cnt_lsb_minus4} + 4)}$$

35 El valor de log2_max_pic_order_cnt_lsb_minus4 estará en el intervalo de 0 a 12, inclusive. sps_max_dec_pic_buffering_minus1 plus 1 especifica el tamaño máximo requerido del búfer de imágenes decodificadas para la CVS en unidades de búferes de almacenamiento de imágenes. El valor de sps_max_dec_pic_buffering_minus1 estará en el intervalo de 0 a MaxDpbSize - 1, inclusive, donde MaxDpbSize es como se especifica en otro lugar. additional_ref_poc_lsb especifica el valor de la variable MaxRefPicOrderCntLsb que se utiliza en el proceso de decodificación para las listas de imágenes de referencia de la siguiente manera:

$$\text{MaxRefPicOrderCntLsb} = 2^{(\log_2 \text{max_pic_order_cnt_lsb_minus4} + 4 + \text{additional_ref_poc_lsb})}$$

45 El valor de additional_ref_poc_lsb estará en el intervalo de 0 a 32 - log2_max_pic_order_cnt_lsb_minus4 - 4, inclusive.

Semántica de RBSP de conjunto de parámetros de imagen.

50 Un num_ref_idx_default_active_minus1[i] plus 1, cuando i es igual a 0, especifica el valor deducido de la variable NumRefIdxActive[0] para las porciones P o B con num_ref_idx_active_override_flag igual a 0 y, cuando i es igual a 1, especifica el valor deducido de NumRefIdxActive[1] para las porciones B con num_ref_idx_active_override_flag igual a 0. El valor de num_ref_idx_default_active_minus1[i] estará en el intervalo de 0 a 14, inclusive.

Semántica de encabezado de porción.

55 Cuando estén presentes, el valor de cada uno de los elementos de sintaxis de encabezado de porción slice_pic_parameter_set_id y slice_pic_order_cnt_lsb será el mismo en todos los encabezados de porción de una imagen codificada.... slice_type especifica el tipo de codificación de la porción de acuerdo con la Tabla 7-3.

Tabla 7-3 - Asociación de nombres a slice_type

slice_type	Nombre de slice_type
0	B (porción B)
1	P (porción P)
2	I (porción I)

Cuando `nal_unit_type` es igual a `IRAP_NUT`, es decir, la imagen es una imagen IRAP, `slice_type` será igual a 2. ... `slice_pic_order_cnt_lsb` especifica el módulo de recuento de orden de imágenes `MaxPicOrderCntLsb` para la imagen actual. La longitud del elemento de sintaxis `slice_pic_order_cnt_lsb` es $\log_2(\text{max_pic_order_cnt_lsb_minus4} + 4)$ bits. El valor de `slice_pic_order_cnt_lsb` estará en el intervalo de 0 a `MaxPicOrderCntLsb - 1`, inclusive. Cuando `slice_pic_order_cnt_lsb` no está presente, se deduce que `slice_pic_order_cnt_lsb` es igual a 0. `num_ref_idx_active_override_flag` igual a 1 especifica que el elemento de sintaxis `num_ref_idx_active_minus1[ 0 ]` está presente para las porciones P y B y que el elemento de sintaxis `num_ref_idx_active_minus1[ 1 ]` está presente para las porciones B. `num_ref_idx_active_override_flag` igual a 0 especifica que los elementos de sintaxis `num_ref_idx_active_minus1[0]` y `num_ref_idx_active_minus1[ 1 ]` no están presentes. `num_ref_idx_active_minus1[ i ]`, cuando está presente, especifica el valor de la variable `NumRefIdxActive[ i ]` de la siguiente manera:

`NumRefIdxActive[ i ] = num_ref_idx_active_minus1[ i ] + 1`

El valor de `num_ref_idx_active_minus1[ i ]` estará en el intervalo de 0 a 14, inclusive. El valor de `NumRefIdxActive[ i ] - 1` especifica el índice de referencia máximo para la lista `i` de imágenes de referencia que se utiliza para decodificar la porción. Cuando el valor de `NumRefIdxActive[ i ]` es igual a 0, no se utiliza ningún índice de referencia para la lista `i` de imágenes de referencia para decodificar la porción. Para `i` igual a 0 o 1, cuando la porción actual es una porción B y `num_ref_idx_active_override_flag` es igual a 0, se deduce que `NumRefIdxActive[ i ]` es igual a `num_ref_idx_default_active_minus1[ i ] + 1`. Cuando la porción actual es una porción P y `num_ref_idx_active_override_flag` es igual a 0, se deduce que `NumRefIdxActive[ 0 ]` es igual a `num_ref_idx_default_active_minus1[ 0 ] + 1`. Cuando la porción actual es una porción P, se deduce que `NumRefIdxActive[ 1 ]` es igual a 0. Cuando la porción actual es una porción I, se deduce que tanto `NumRefIdxActive[ 0 ]` como `NumRefIdxActive[ 1 ]` son iguales a 0. Como alternativa, para `i` igual a 0 o 1, se aplica lo siguiente después de lo anterior: sea `rplIdx1` establecido como `igual a ref_pic_list_sps_flag[ i ] ? ref_pic_list_idx[ i ] : num_ref_pic_lists_in_sps[ i ]`, y sea `numRpEntries[ i ]` igual a `num_strp_entries[ i ][ rplIdx1 ] + num_ltrp_entries[ i ][ rplIdx1 ]`. Cuando `NumRefIdxActive[ i ]` es mayor que `numRpEntries[ i ]`, el valor de `NumRefIdxActive[ i ]` se establece igual a `numRpEntries[ i ]`.

Semántica de estructura de lista de imágenes de referencia.

La estructura de sintaxis `ref_pic_list_struct( listIdx )` está presente en un encabezado de porción. Cuando está presente en un encabezado de porción, la estructura de sintaxis `ref_pic_list_struct( listIdx )` especifica la lista `listIdx` de imágenes de referencia de la imagen actual (la imagen que contiene la porción). `num_ref_entries[ listIdx ]` especifica el número de entradas en la estructura de sintaxis `ref_pic_list_struct( listIdx )`. La variable `NumEntriesInList[ listIdx ]` se obtiene de la siguiente manera:

`NumRefPicEntriesInRpl[ listIdx ] = num_ref_entries[ listIdx ]`

El valor de `NumRefPicEntries[ listIdx ]` estará en el intervalo de 0 a `sps_max_dec_pic_buffering_minus1`, inclusive. `poc_ref_lsb[ listIdx ][ i ]` especifica el valor del módulo de recuento de orden de imágenes `MaxRefPicOrderCntLsb` de la imagen a la que hace referencia la `i`-ésima entrada en la estructura de sintaxis `ref_pic_list_struct( listIdx )`. La longitud del elemento de sintaxis `poc_ref_lsb[ listIdx ][ i ]` es $\log_2(\text{MaxRefPicOrderCntLsb})$ bits.

Se expone el proceso de decodificación.

Proceso general de decodificación.

El proceso de decodificación funciona de la siguiente manera para la imagen actual `CurrPic`: la decodificación de unidades NAL se especifica posteriormente. Los procesos expuestos posteriormente especifican los siguientes procesos de decodificación utilizando elementos de sintaxis en la capa de encabezado de porción y superiores: se obtienen variables y funciones en relación con el recuento de orden de imágenes. Es necesario invocar esto sólo para la primera porción de una imagen. Al comienzo del proceso de decodificación para cada porción de una imagen no-IRAP, se invoca el proceso de decodificación para la construcción de listas de imágenes de referencia para la obtención de la lista 0 de imágenes de referencia (`RefPicList[ 0 ]`) y la lista 1 de imágenes de referencia (`RefPicList[ 1 ]`). Se invoca el proceso de decodificación para el marcado de imágenes de referencia, en donde las imágenes de referencia se marcan como "no utilizadas para referencia". Es necesario invocar esto sólo para la primera porción de una imagen. Se invocan los procesos de decodificación para codificar unidades de árbol, cambiar la escala, transformar, filtrar en bucle, etc. Después de que se hayan decodificado todas las porciones de la imagen actual, la imagen decodificada actual se marca como "utilizada para referencia".

Proceso de decodificación de unidades NAL.

Las entradas a este proceso son unidades NAL de la imagen actual y sus unidades NAL no-VCL asociadas. Las salidas de este proceso son las estructuras de sintaxis de RBSP analizadas encapsuladas dentro de las unidades NAL. El proceso de decodificación para cada unidad NAL extrae la estructura de sintaxis de RBSP de la unidad NAL y, a continuación, analiza la estructura de sintaxis de RBSP.

Proceso de decodificación de porciones.

Proceso de decodificación para el recuento de orden de imágenes.

La salida de este proceso es `PicOrderCntVal`, el recuento de orden de imágenes de la imagen actual. Los recuentos de orden de imágenes se utilizan para identificar imágenes, para obtener parámetros de movimiento en modo de fusión y la predicción de vectores de movimiento, y para la verificación de conformidad del decodificador. Cada imagen codificada está asociada con una variable de recuento de orden de imágenes, denominada `PicOrderCntVal`. Cuando la imagen actual no es una imagen IRAP, las variables `prevPicOrderCntLsb` y `prevPicOrderCntMsb` se obtienen de la siguiente manera: sea `prevTid0Pic` la imagen anterior en orden de decodificación que tenga `TemporalId` igual a 0. La variable `prevPicOrderCntLsb` se establece como igual a `slice_pic_order_cnt_lsb` de `prevTid0Pic`. La variable `prevPicOrderCntMsb` se establece como igual a `PicOrderCntMsb` de `prevTid0Pic`. La variable `PicOrderCntMsb` de la imagen actual se obtiene de la siguiente manera: si la imagen actual es una imagen IRAP, `PicOrderCntMsb` se establece como igual a 0. De lo contrario, `PicOrderCntMsb` se obtiene de la siguiente manera:

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

`PicOrderCntVal` se obtiene de la siguiente manera:

```
PicOrderCntVal = PicOrderCntMsb + slice_pic_order_cnt_lsb
```

Todas las imágenes IRAP tendrán `PicOrderCntVal` igual a 0, ya que se deduce que `slice_pic_order_cnt_lsb` es 0 para las imágenes IRAP y `prevPicOrderCntLsb` y `prevPicOrderCntMsb` se establecen ambos como iguales a 0. El valor de `PicOrderCntVal` estará en el intervalo de -231 a 231 - 1, inclusive. En una CVS, los valores de `PicOrderCntVal` para dos imágenes codificadas cualesquiera no serán los mismos. En cualquier momento durante el proceso de decodificación, los valores de `PicOrderCntVal` & (`MaxRefPicOrderCntLsb` - 1) para dos imágenes de referencia cualesquiera en el DPB no serán los mismos.

La función `PicOrderCnt( picX )` se especifica de la siguiente manera:

```
PicOrderCnt( picX ) = PicOrderCntVal de la imagen picX
```

La función `DiffPicOrderCnt( picA, picB )` se especifica de la siguiente manera:

```
DiffPicOrderCnt( picA, picB ) = PicOrderCnt( picA ) - PicOrderCnt( picB )
```

El flujo de bits no contendrá datos que den como resultado valores de `DiffPicOrderCnt( picA, picB )` utilizados en el proceso de decodificación que no estén en el intervalo de -215 a 215 - 1, inclusive. Sea `X` la imagen actual y sean `Y` y `Z` otras dos imágenes en la misma CVS, `Y` y `Z` se considera que están en la misma dirección de orden de salida de `X` cuando `DiffPicOrderCnt( X, Y )` y `DiffPicOrderCnt( X, Z )` son ambos positivos o ambos negativos.

Proceso de decodificación para la construcción de listas de imágenes de referencia.

Este proceso se invoca al principio del proceso de decodificación para cada porción de una imagen no-IRAP. Las imágenes de referencia se direccionan a través de índices de referencia. Un índice de referencia es un índice en una lista de imágenes de referencia. Al decodificar una porción `I`, no se utiliza ninguna lista de imágenes de referencia en la decodificación de los datos de la porción. Al decodificar una porción `P`, sólo se utiliza la lista 0 de imágenes de referencia (es decir, `RefPicList[ 0 ]`) en la decodificación de los datos de la porción. Al decodificar una porción `B`, tanto la lista 0 de imágenes de referencia como la lista 1 de imágenes de referencia (es decir, `RefPicList[ 1 ]`) se utilizan en la decodificación de los datos de la porción. Al comienzo del proceso de decodificación para cada porción de una imagen no-IRAP, se obtienen las listas `RefPicList[ 0 ]` y `RefPicList[ 1 ]` de imágenes de referencia. Las listas de

imágenes de referencia se utilizan en el marcado de imágenes de referencia o en la decodificación de los datos de la porción. Para una porción I de una imagen no-IRAP que no sea la primera porción de la imagen, RefPicList[0] y RefPicList[1] se obtienen con el propósito de verificar la conformidad del flujo de bits, pero su obtención no es necesaria para decodificar la o las imágenes actuales que siguen a la imagen actual en orden de decodificación. Para una porción P que no sea la primera porción de una imagen, RefPicList[1] se obtiene con el propósito de verificar la conformidad del flujo de bits, pero su obtención no es necesaria para decodificar la o las imágenes actuales que siguen a la imagen actual en orden de decodificación. Las listas RefPicList[0] y RefPicList[1] de imágenes de referencia se construyen de la siguiente manera:

```

for( i = 0; i < 2; i++ ) {
    for( j = 0; picOrderCntVal < NumEntriesInList[ i ]; j++ ) {
        if( hay una picA de referencia en el DPB con PicOrderCntVal & ( MaxRefPicOrderCntLsb - 1 )
            igual a pic_ref_list[ i ][ j ] )
            RefPicList[ i ][ j ] = picA;
        else
            RefPicList[ i ][ j ] = "ninguna imagen de referencia";
    }
}

```

Para cada i igual a 0 o 1, las primeras entradas NumRefIdxActive[i] en RefPicList[i] se denominan entradas activas en RefPicList[i], y las otras entradas en RefPicList[i] se denominan entradas inactivas en RefPicList[i]. Es posible que a una imagen en particular hagan referencia tanto una entrada en RefPicList[0] como una entrada en RefPicList[1]. También es posible que a una imagen en particular hagan referencia más de una entrada en RefPicList[0] o más de una entrada en RefPicList[1]. Las entradas activas en RefPicList[0] y las entradas activas en RefPicList[1] se refieren colectivamente a todas las imágenes de referencia que se utilizan para la interpretación de la imagen actual y una o más imágenes que siguen a la imagen actual en orden de decodificación. Las entradas inactivas en RefPicList[0] y las entradas inactivas en RefPicList[1] se refieren colectivamente a todas las imágenes de referencia que no se utilizan para la interpretación de la imagen actual, pero se utilizan para la interpretación de una o más imágenes que siguen a la imagen actual en orden de decodificación. Hay una o más entradas en RefPicList[0] o RefPicList[1] que son iguales a "ninguna imagen de referencia" porque las imágenes correspondientes no están presentes en el DPB. Cada entrada inactiva en RefPicList[0] o RefPicList[1] que sea igual a "ninguna imagen de referencia" debe ignorarse. Se debe deducir una pérdida de imagen no intencionada para cada entrada activa en RefPicList[0] o RefPicList[1] que sea igual a "ninguna imagen de referencia".

Es un requisito de conformidad de flujo de bits que se apliquen las siguientes restricciones: para cada i igual a 0 o 1, NumEntriesInList[i] no será menor que NumRefIdxActive[i]. La imagen a la que haga referencia cada entrada activa en RefPicList[0] o RefPicList[1] estará presente en el DPB y tendrá TemporalId menor o igual que el de la imagen actual. Opcionalmente, se especifica además la siguiente restricción: el índice de entrada de cualquier entrada inactiva en RefPicList[0] o RefPicList[1] no se utilizará como índice de referencia para la decodificación de la imagen actual. Opcionalmente, se especifica además la siguiente restricción: una entrada inactiva en RefPicList[0] o RefPicList[1] no se referirá a la misma imagen que cualquier otra entrada en RefPicList[0] o RefPicList[1]. A la imagen actual en sí misma no hará referencia ninguna entrada en RefPicList[0] o RefPicList[1]. No habrá ninguna entrada en RefPicList[0] o RefPicList[1] para la que la diferencia entre PicOrderCntVal de la imagen actual y PicOrderCntVal de la imagen a la que hace referencia la entrada sea superior o igual a 224. Sea setOfRefPics el conjunto de imágenes únicas a las que hacen referencia todas las entradas de RefPicList[0] y todas las entradas de RefPicList[1]. El número de imágenes en setOfRefPics será menor o igual que sps_max_dec_pic_buffering_minus1, y setOfRefPics será el mismo para todas las porciones de una imagen.

Proceso de decodificación para el marcado de imágenes de referencia.

Este proceso se invoca una vez por imagen, después de decodificar un encabezado de porción y el proceso de decodificación para la construcción de listas de imágenes de referencia para la porción, pero antes de la decodificación de los datos de la porción. Este proceso da como resultado que una o más imágenes de referencia en el DPB se marquen como "no utilizadas para referencia". Una imagen decodificada en el DPB puede marcarse como "no utilizada para referencia" o "utilizada para referencia", pero sólo una de estas dos en cualquier momento dado durante la operación del proceso de decodificación. Asignar una de estas marcas a una imagen elimina implícitamente otra de estas marcas cuando corresponda. Cuando la imagen actual es una imagen IRAP, todas las imágenes de referencia actualmente presentes en el DPB (si las hay) se marcan como "no utilizadas para referencia". Las imágenes de referencia en el DPB se identifican mediante los $\text{Log}_2(\text{MaxRefPicOrderCntLsb})$ LSB de sus valores PicOrderCntVal. Cada imagen de referencia en el DPB a la que no haga referencia ninguna entrada de RefPicList[0] o RefPicList[1] está marcada como "no utilizada para referencia".

Otra realización alternativa más.

Esta sección describe una realización alternativa al enfoque especificado anteriormente que se denomina

"señalización siempre de listas de imágenes de referencia en encabezados de porción con diferenciación entre imágenes de referencia a corto y largo plazo". En esta realización alternativa, en el encabezado de porción, se señala un ciclo de MSB de POC para cada entrada LTRP, de manera similar a como se hace en HEVC o en los enfoques descritos anteriormente, y se elimina la siguiente restricción: en cualquier momento durante el proceso de decodificación, los valores de $\text{PicOrderCntVal} \& (\text{MaxLtPicOrderCntLsb} - 1)$ para dos imágenes de referencia cualesquiera en el DPB no serán los mismos.

La Figura 6 es un diagrama esquemático de un dispositivo 600 de codificación de vídeo (por ejemplo, un codificador 20 de vídeo o un decodificador 30 de vídeo) según una realización de la divulgación. El dispositivo 600 de codificación de vídeo es adecuado para implementar las realizaciones divulgadas como se describe en la presente memoria. El dispositivo 600 de codificación de vídeo comprende puertos 610 de entrada y unidades receptoras (Rx) 620 para recibir datos; un procesador, unidad lógica o unidad central 630 de procesamiento (CPU, por sus siglas en inglés) para procesar los datos; unidades transmisoras (Tx) 640 y puertos 650 de salida para transmitir los datos; y una memoria 660 para almacenar los datos. El dispositivo 600 de codificación de vídeo también comprende componentes óptico a eléctrico (OE) y componentes eléctrico a óptico (EO) acoplados a los puertos 610 de entrada, las unidades receptoras 620, las unidades transmisoras 640 y los puertos 650 de salida para la salida o entrada de señales ópticas o eléctricas.

El procesador 630 está implementado mediante *hardware* y *software*. El procesador 630 se implementa como uno o más chips de CPU, núcleos (por ejemplo, como un procesador multinúcleo), agrupaciones de puertas programables de campo (FPGA, por sus siglas en inglés), circuitos integrados de aplicación específica (ASIC, por sus siglas en inglés) y procesadores de señales digitales (DSP, por sus siglas en inglés). El procesador 630 está en comunicación con los puertos 610 de entrada, las unidades receptoras 620, las unidades transmisoras 640, los puertos 650 de salida y la memoria 660. El procesador 630 comprende un módulo 670 de codificación. El módulo 670 de codificación implementa las realizaciones divulgadas descritas anteriormente. Por ejemplo, el módulo 670 de codificación implementa, procesa, prepara o proporciona las diversas funciones de conexión en red. La inclusión del módulo 670 de codificación por lo tanto proporciona una mejora sustancial de la funcionalidad del dispositivo 600 de codificación de vídeo y efectúa una transformación del dispositivo 600 de codificación de vídeo en un estado diferente. Como alternativa, el módulo 670 de codificación se implementa como instrucciones almacenadas en la memoria 660 y ejecutadas por el procesador 630.

El dispositivo 600 de codificación de vídeo también incluye dispositivos 680 de entrada y/o salida (E/S) para comunicar datos a y de un usuario. Los dispositivos 680 de E/S incluyen dispositivos de salida, tales como una pantalla para visualizar datos de vídeo, altavoces para emitir datos de audio, etc. Los dispositivos 680 de E/S también incluyen dispositivos de entrada, tales como un teclado, ratón, bola rastreadora, etc., y/o interfaces correspondientes para interactuar con tales dispositivos de salida.

La memoria 660 comprende uno o más discos, unidades de cinta magnética y/o unidades de estado sólido y se usa como un dispositivo de almacenamiento de datos de desbordamiento para almacenar programas, cuando tales programas se seleccionan para ejecutarlos, y para almacenar instrucciones y datos que se leen durante la ejecución del programa. La memoria 660 es volátil y/o no volátil y es memoria de sólo lectura (ROM, por sus siglas en inglés), memoria de acceso aleatorio (RAM, por sus siglas en inglés), memoria ternaria de contenido direccionable (TCAM, por sus siglas en inglés) y/o memoria estática de acceso aleatorio (SRAM, por sus siglas en inglés).

La Figura 7 es un diagrama esquemático de una realización de un medio 700 de codificación. En una realización, el medio 700 de codificación se implementa en un dispositivo 702 de codificación de vídeo (por ejemplo, un codificador 20 de vídeo o un decodificador 30 de vídeo). El dispositivo 702 de codificación de vídeo incluye medios receptores 701. Los medios receptores 701 están configurados para recibir una imagen para codificar o para recibir un flujo de bits para decodificar. El dispositivo 702 de codificación de vídeo incluye medios transmisores 707 acoplados a los medios receptores 701. Los medios transmisores 707 están configurados para transmitir el flujo de bits a un decodificador o para transmitir una imagen decodificada a un medio de visualización (por ejemplo, uno de los dispositivos 680 de E/S).

El dispositivo 702 de codificación de vídeo incluye unos medios 703 de almacenamiento. Los medios 703 de almacenamiento están acoplados a al menos uno de los medios receptores 701 o los medios transmisores 707. Los medios 703 de almacenamiento están configurados para almacenar instrucciones. El dispositivo 702 de codificación de vídeo también incluye medios 705 de procesamiento. Los medios 705 de procesamiento están acoplados a los medios 703 de almacenamiento. Los medios 705 de procesamiento están configurados para ejecutar las instrucciones almacenadas en los medios 703 de almacenamiento para realizar los métodos divulgados en la presente memoria.

REIVINDICACIONES

1. Un método para decodificar un flujo de bits de vídeo codificado implementado mediante un decodificador de vídeo basado en el estándar VVC, en donde el flujo de bits de vídeo codificado comprende un encabezado de porción de una porción actual y datos que representan la porción actual, comprendiendo el método:
5 analizar el encabezado de porción del flujo de bits de vídeo codificado, en donde el encabezado de porción comprende una estructura de lista de imágenes de referencia;
10 obtener, sobre la base de la estructura de lista de imágenes de referencia, dos listas de imágenes de referencia de la porción actual, en donde, en cada una de las dos listas de imágenes de referencia, sólo cierto número de entradas activas al principio de cada una de las dos listas de imágenes de referencia hacen referencia a las imágenes de referencia que se utilizan para la interpredicción de la imagen actual, mientras que otras entradas de la lista se denominan entradas inactivas, las entradas inactivas hacen referencia a todas las imágenes de referencia que no se
15 utilizan para la interpredicción de la imagen actual, pero se utilizan en la interpredicción para una o más imágenes que siguen a la imagen actual en orden de decodificación; y
obtener, basándose en la lista de imágenes de referencia, al menos un bloque reconstruido de la porción actual.
20 2. El método de la reivindicación 1, en donde un orden de entradas en las estructuras de lista de imágenes de referencia es el mismo que un orden de imágenes de referencia correspondientes en la lista de imágenes de referencia, en donde cada entrada en las estructuras de lista de imágenes de referencia describe una imagen de referencia correspondiente en la lista de imágenes de referencia.
25 3. El método de cualquiera de las reivindicaciones 1 a 2, en donde el al menos un bloque reconstruido se utiliza para generar una imagen visualizada en una pantalla de un dispositivo electrónico.
4. El método de cualquiera de las reivindicaciones 1 a 3, en donde la lista de imágenes de referencia comprende una lista de imágenes de referencia utilizadas para la interpredicción del al menos un bloque reconstruido.
30 5. El método de cualquiera de las reivindicaciones 1 a 4, en donde la porción actual es una porción P o para una porción B.
6. Un decodificador que comprende circuitería de procesamiento para llevar a cabo el método según cualquiera de las
35 reivindicaciones 1 a 5.
7. Un producto de programa informático que comprende un código de programa para realizar el método según cualquiera de las reivindicaciones 1 a 5 cuando se ejecuta en un ordenador o un procesador.

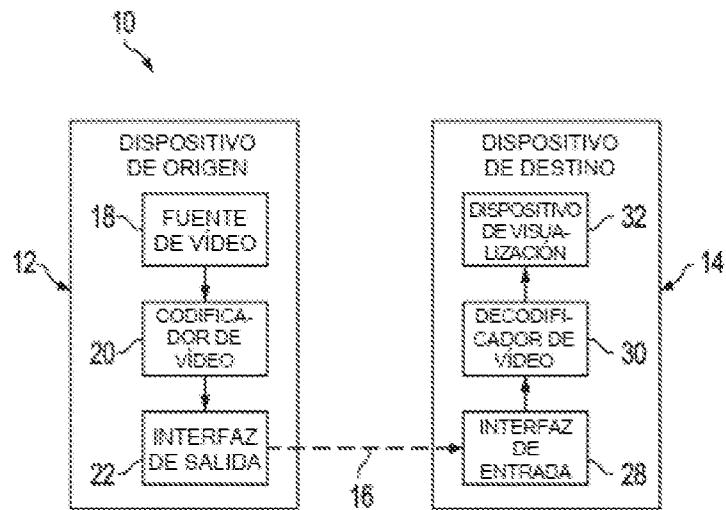


FIG. 1

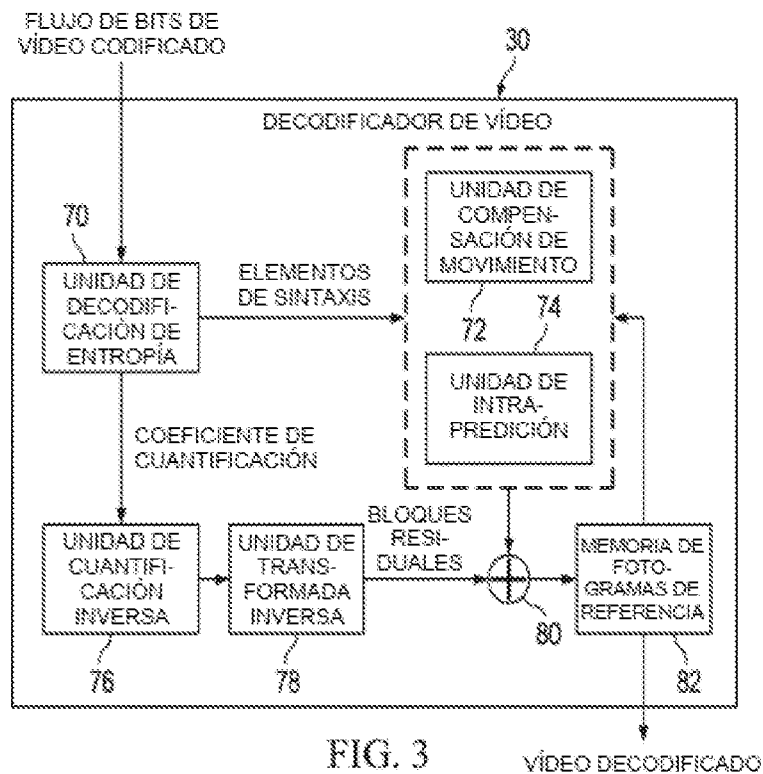


FIG. 3

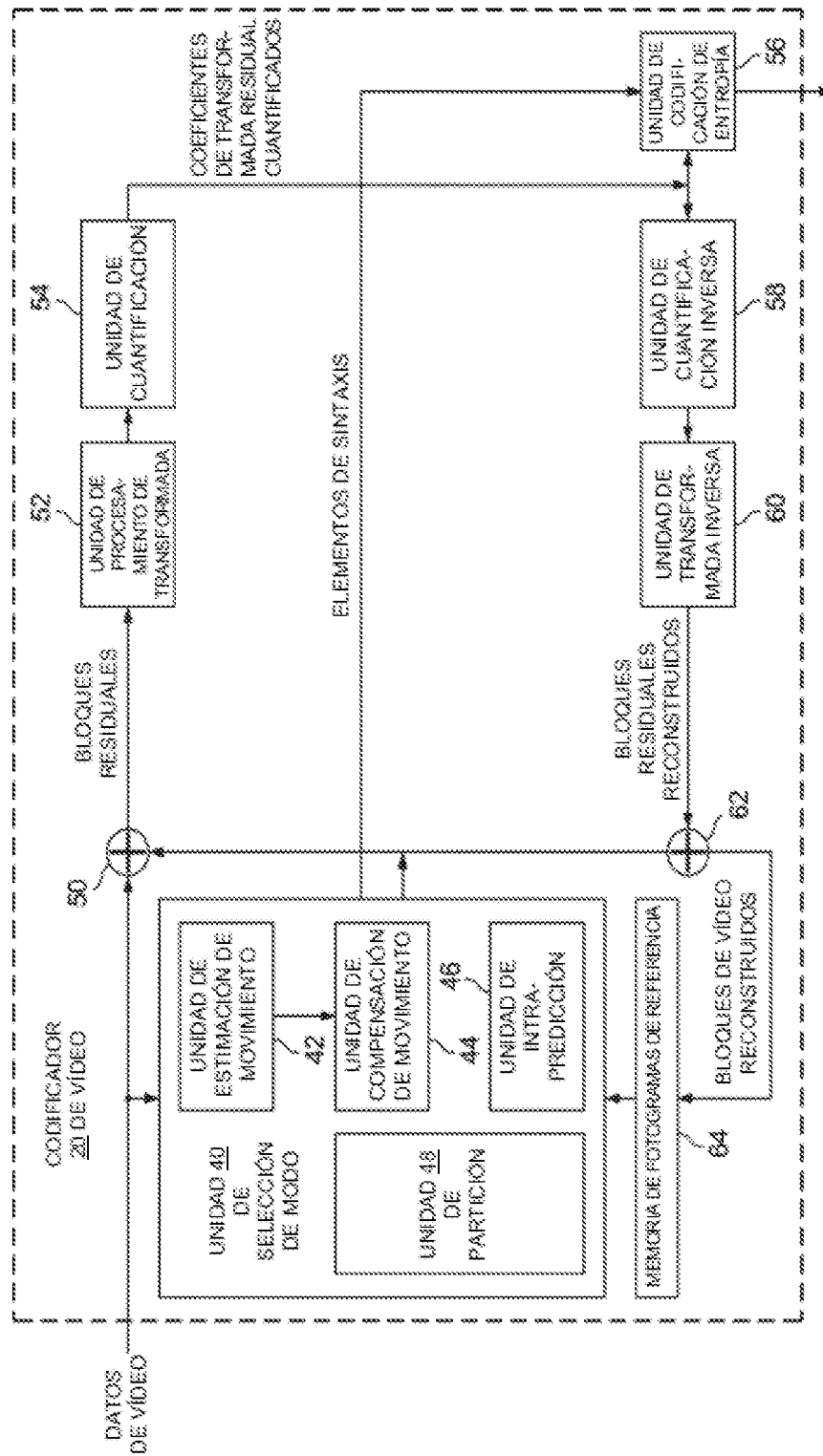


Fig. 2

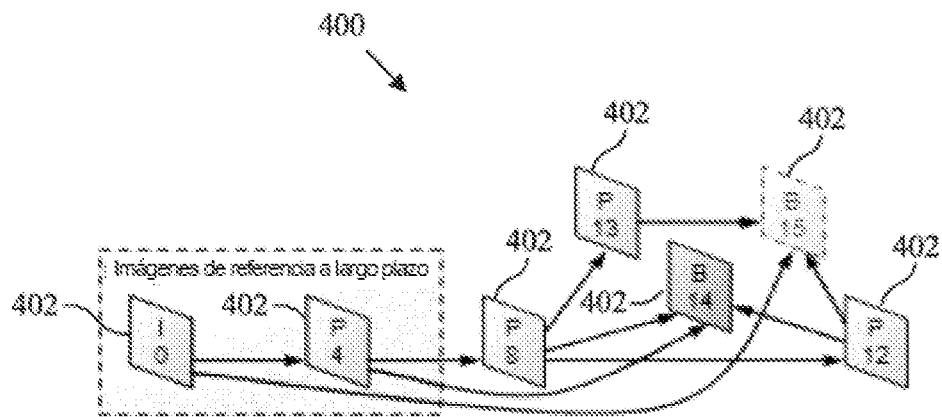


FIG. 4

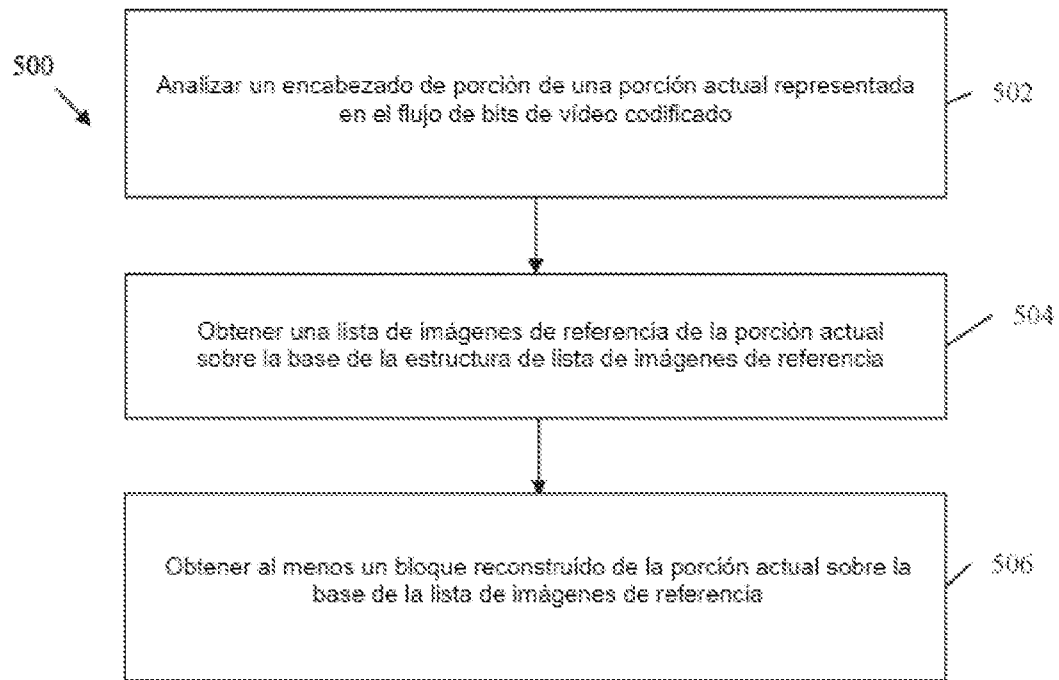


FIG. 5

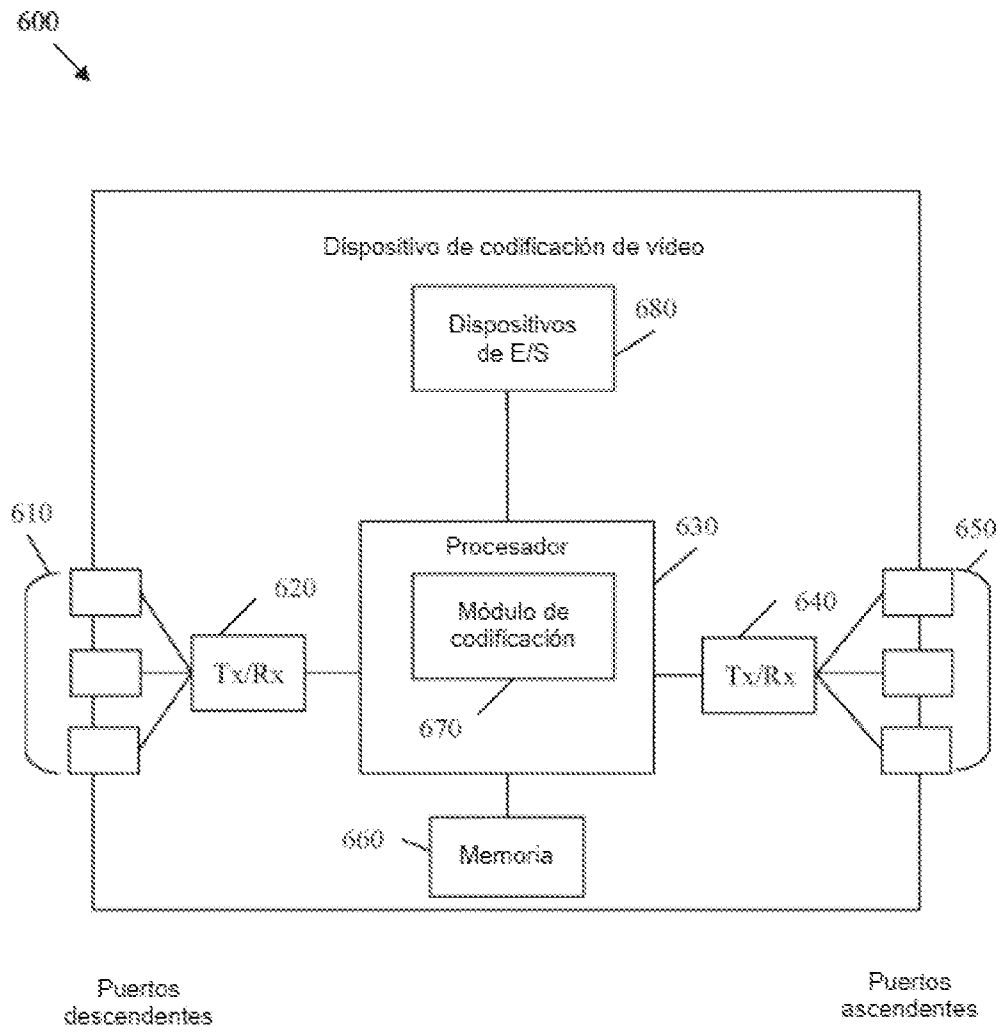


FIG. 6

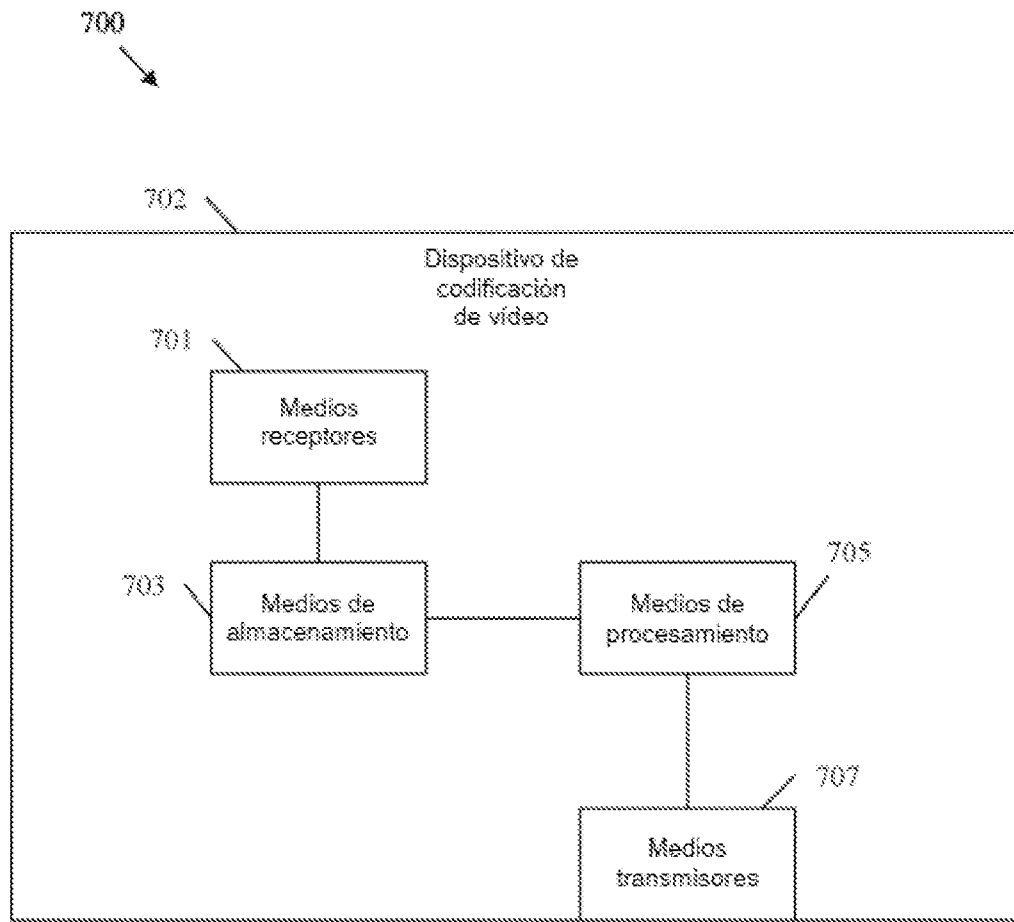


FIG. 7