

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 981 234**

51 Int. Cl.:

**H04N 19/70** (2014.01)

**H04N 19/58** (2014.01)

**H04N 19/105** (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **12.09.2019 PCT/US2019/050861**

87 Fecha y número de publicación internacional: **19.03.2020 WO20056172**

96 Fecha de presentación y número de la solicitud europea: **12.09.2019 E 19858808 (9)**

97 Fecha y número de publicación de la concesión europea: **14.02.2024 EP 3847812**

54 Título: **Candidato de señalización para estructuras de listas de imágenes de referencia**

30 Prioridad:

**12.09.2018 US 201862730172 P**

**15.05.2019 US 201962848147 P**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:  
**07.10.2024**

73 Titular/es:

**HUAWEI TECHNOLOGIES CO., LTD. (100.0%)  
Huawei Administration Building Bantian,  
Longgang District  
Shenzhen, Guangdong 518129, CN**

72 Inventor/es:

**HENDRY, FNU y  
WANG, YE-KUI**

74 Agente/Representante:

**PONS ARIÑO, Ángel**

ES 2 981 234 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**

Candidato de señalización para estructuras de listas de imágenes de referencia

Campo de la invención

5 En general, esta descripción describe técnicas para señalar mejoras de eficiencia en la gestión de imágenes de referencia en la codificación de video. De manera más específica, esta descripción describe técnicas para señalización mejorada para la construcción de listas de imágenes de referencia y la marcación de imágenes de referencia que se basa directamente en las listas de imágenes de referencia.

Antecedentes

10 La cantidad de datos de video necesarios para mostrar incluso un video relativamente corto puede ser sustancial, lo cual puede dar como resultado dificultades cuando los datos han de ser transmitidos o comunicados de otra manera a través de una red de comunicaciones con capacidad de ancho de banda limitada. Así, los datos de video generalmente son comprimidos antes de ser comunicados a través de redes de telecomunicaciones por módem de hoy en día. El tamaño de un video también podría ser un problema cuando el video es almacenado en un dispositivo de almacenamiento debido a que los recursos de memoria pueden ser limitados. Los dispositivos de compresión de video con frecuencia utilizan software y/o hardware en la fuente para codificar los datos de video antes de la transmisión o almacenamiento, disminuyendo así la cantidad de datos necesarios para representar imágenes de video digitales. Los datos comprimidos entonces son recibidos en el destino por un dispositivo de descompresión de video que decodifica los datos de video. Con recursos de red limitados y demandas cada vez mayores de calidad de video superior, son deseables técnicas de compresión y descompresión mejoradas que mejoren la relación de compresión con de poco a ningún sacrificio en la calidad de la imagen. La solicitud de patente PCT n.º WO2012067966A1 describe un proceso general de decodificación para crear una segunda lista de imágenes de referencia como duplicado de una primera lista de imágenes de referencia. La solicitud de patente de EE. UU. n.º US 2018/199051 A1 describe un proceso de decodificación para generar una lista de imágenes de referencia basada en las estructuras sintácticas de descripción de la lista de imágenes de referencia.

Compendio

30 La presente invención se define en las reivindicaciones independientes. La habilitación de la descripción de la invención protegida es proporcionada con las realizaciones descritas en relación con la sintaxis del conjunto de los parámetros de secuencia, RBSP y la semántica correspondiente. Las demás figuras, aspectos y realizaciones se proporcionan con fines ilustrativos y no representan realizaciones de la invención a menos que se combinen con todas las características definidas respectivamente en las reivindicaciones independientes.

Breve descripción de los dibujos

35 Para una comprensión más completa de esta descripción, ahora se hace referencia a la siguiente breve descripción, tomada en combinación con los dibujos adjuntos y la descripción detallada, en donde números de referencia similares representan partes similares.

La figura 1 es un diagrama de bloques que ilustra un sistema de codificación ejemplar que puede utilizar técnicas de predicción bilaterales.

40 La figura 2 es un diagrama de bloques que ilustra un codificador de video ejemplar que puede implementar técnicas de predicción bilaterales.

La figura 3 es un diagrama de bloques que ilustra un ejemplo de un decodificador de video que puede implementar técnicas de predicción bilaterales.

La figura 4 es un diagrama esquemático que ilustra un conjunto de imágenes de referencia (RPS) que tiene una imagen actual con entradas en todos los subconjuntos del RPS.

45 La figura 5 es un diagrama esquemático de una realización de una corriente de bits de video.

La figura 6 es una realización de un método de decodificación de una corriente de bits de video codificada.

La figura 7 es una realización de un método de codificación de una corriente de bits de video.

La figura 8 es un diagrama esquemático de un dispositivo de codificación de video.

La figura 9 es un diagrama esquemático de una realización de un medio para codificación.

50

## Descripción detallada

Los siguientes son diversos acrónimos empleados en la presente memoria: Memoria Tampón de Imágenes Decodificadas (DPB), Actualizar Decodificación Instantánea (IDR), Punto de Acceso Intra-Aleatorio (IRAP), Bit Menos Significativo (LSB), Bit Más Significativo (MSB), Capa de Abstracción de Red (NAL), Recuento de Orden de Imágenes (POC), Carga Útil de Secuencia de Bytes sin Procesar (RBSP), Conjunto de Parámetros de Secuencia (SPS), y Borrador de Trabajo (WD).

La figura 1 es un diagrama de bloques que ilustra un sistema 10 de codificación ejemplar que puede utilizar técnicas de codificación de video como se describe en la presente memoria. Como se muestra en la figura 1, el sistema 10 de codificación incluye un dispositivo fuente 12 que proporciona datos de video codificados que han de ser decodificados en un momento posterior por un dispositivo destino 14. En particular, el dispositivo fuente 12 puede proporcionar los datos de video al dispositivo destino 14 mediante un medio 16 legible por ordenador. El dispositivo fuente 12 y el dispositivo destino 14 pueden comprender cualquiera de una amplia variedad de dispositivos, incluyendo ordenadores de escritorio, ordenadores portátiles (por ejemplo, "laptop"), ordenadores de tableta, decodificadores, equipos telefónicos tales como los denominados teléfonos "inteligentes", los denominados teclados "inteligentes", televisiones, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de juegos de video, dispositivo de transmisión de video, o similar. En algunos casos, el dispositivo fuente 12 y el dispositivo destino 14 pueden estar equipados para comunicación inalámbrica.

El dispositivo destino 14 puede recibir los datos de video codificados que han de ser decodificados a través del medio 16 legible por ordenador. El medio 16 legible por ordenador puede comprender cualquier tipo de medio o dispositivo capaz de mover los datos de video codificados desde el dispositivo fuente 12 al dispositivo destino 14. En un ejemplo, el medio 16 legible por ordenador puede comprender un medio de comunicación para permitir que el dispositivo fuente 12 transmita datos de video codificados directamente al dispositivo destino 14 en tiempo real. Los datos de video codificados pueden ser modulados según una norma de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitidos al dispositivo destino 14. El medio de comunicación puede comprender cualquier medio de comunicación cableado o inalámbrico, tal como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión física. El medio de comunicación puede formar parte de una red basada en paquetes, tal como una red de área local, una red de área amplia, o una red global tal como la Internet. El medio de comunicación puede incluir rúteres, interruptores, estaciones base, o cualquier otro equipo que puede ser útil para facilitar la comunicación desde el dispositivo fuente 12 al dispositivo destino 14.

En algunos ejemplos, datos codificados pueden ser emitidos desde la interfaz 22 de salida a un dispositivo de almacenamiento. De manera similar, se puede tener acceso a datos codificados desde el dispositivo de almacenamiento por la interfaz de entrada. El dispositivo de almacenamiento puede incluir cualquiera de una variedad de medios de almacenamiento de datos distribuidos o a los que se tiene acceso localmente tal como un disco duro, discos Blu-ray, discos de video digital (DVD), Memorias de Solo 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 video codificados. En un ejemplo adicional, el dispositivo de almacenamiento puede corresponder a un servidor de archivos u otro dispositivo de almacenamiento intermedio que puede almacenar el video codificado generado por el dispositivo fuente 12. El dispositivo destino 14 puede tener acceso a datos de video almacenados desde el dispositivo de almacenamiento a mediante transmisión o descarga. El servidor de archivos puede ser cualquier tipo de servidor capaz de almacenar datos de video codificados y transmitir esos datos de video codificados al dispositivo destino 14. Servidores de archivos ejemplares incluyen un servidor web (por ejemplo, para un sitio web), un servidor de protocolo de transferencia de archivos (FTP), dispositivos de almacenamiento unidos a la red (NAS), o una unidad de disco local. El dispositivo destino 14 puede tener acceso a los datos de video codificados a través de cualquier conexión de datos estándar, incluyendo una conexión de Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión Wi-Fi), una conexión cableada (por ejemplo, línea de suscriptor digital (DSL), modem de cable, etc.), o una combinación de ambos que sea adecuada para tener acceso a datos de video codificados almacenados en un servidor de archivos. La transmisión de datos de video codificados desde el dispositivo de almacenamiento puede ser una transmisión por secuencia, una transmisión por descarga, o una combinación de las mismas.

Las técnicas de esta descripción no se limitan necesariamente a aplicaciones o configuraciones inalámbricas. Las técnicas se pueden aplicar a la codificación de video en apoyo de cualquiera de una variedad de aplicaciones multimedia, tal como transmisiones de televisión sobre-el-aire, transmisiones de televisión por cable, transmisiones de televisión satelital, transmisiones de video por secuencias de Internet, tal como transmisión adaptable dinámica sobre HTTP (DASH), video digital que es codificado sobre un medio de almacenamiento de datos, decodificación de video digital almacenado en un medio de almacenamiento de datos u otras aplicaciones. En algunos ejemplos, el sistema 10 de codificación puede ser configurado para soportar transmisión de video de un sentido o de dos sentidos para soportar aplicaciones tales como transmisión por secuencias de video, reproducción de video, difusión de video y/o telefonía por video.

En el ejemplo de la figura 1, el dispositivo fuente 12 incluye la fuente 18 de video, el codificador 20 de video, y la interfaz 22 de salida. El dispositivo destino 14 incluye la interfaz 28 de entrada, el decodificador 30 de video, y el dispositivo 32 de despliegue. Según esta descripción, el codificador 20 de video del dispositivo fuente 12 y/o el decodificador 30 de video del dispositivo destino 14 pueden ser configurados para aplicar las técnicas para la codificación de video. En otros ejemplos, un dispositivo fuente y un dispositivo destino pueden incluir otros componentes o disposiciones. Por ejemplo, el dispositivo fuente 12 puede recibir datos de video procedentes de una fuente de video externa, tal como una cámara externa. De igual forma, el dispositivo destino 14 puede conectarse 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. Técnicas para la codificación de video pueden ser realizadas por cualquier dispositivo de codificación y/o de decodificación de video digital. Aunque las técnicas de esta descripción generalmente son realizadas por un dispositivo de codificación de video, las técnicas también pueden ser realizadas por un codificador/decodificador de video, típicamente denominada como un "CODEC." Además, las técnicas de esta descripción también pueden ser realizadas por un preprocesador de video. El codificador y/o decodificador de video pueden ser una unidad de procesamiento de gráficos (GPU) o un dispositivo similar.

El dispositivo fuente 12 y el dispositivo destino 14 son simplemente ejemplos de dichos dispositivos de codificación en los cuales el dispositivo fuente 12 genera datos de video codificados para transmisión al dispositivo destino 14. En algunos ejemplos, el dispositivo fuente 12 y el dispositivo destino 14 pueden operar de una manera sustancialmente simétrica de tal modo que cada uno de los dispositivos 12, 14 fuente y destino incluye componentes de codificación y decodificación de video. Por lo tanto, el sistema 10 de codificación puede soportar transmisión de video de un sentido o de dos sentidos entre dispositivos 12, 14 de video, por ejemplo, para transmisión de video, reproducción de video, difusión de video, o telefonía de video.

La fuente 18 de video del dispositivo fuente 12 puede incluir un dispositivo de captura de video, tal como una cámara de video, un archivo de video que contiene video previamente capturado, y/o una interfaz de alimentación de video para recibir video procedente de un proveedor de contenido de video. Como una alternativa adicional, la fuente 18 de video puede generar datos basados en gráficos de ordenador como el video fuente, o una combinación de video en vivo, video archivado, y video generado por ordenador.

En algunos casos, cuando la fuente 18 de video es una cámara de video, el dispositivo fuente 12 y el dispositivo destino 14 pueden formar los denominados teléfonos con cámara o teléfonos de video. Tal como se mencionó antes, no obstante, las técnicas descritas en esta descripción pueden ser aplicables a codificación de video en general, y pueden ser aplicadas a aplicaciones inalámbricas y/o cableadas. En cada caso, el video capturado, pre-capturado o generado por ordenador puede ser codificado por el codificador 20 de video. La información de video codificada puede entonces ser emitida por la interfaz 22 de salida sobre un medio 16 legible por ordenador.

El medio 16 legible por ordenador puede incluir medios transitorios, tal como una difusión inalámbrica o transmisión de red cableada, o medio de almacenamiento (es decir, medio de almacenamiento no-transitorio), tal como un disco duro, unidad flash, disco compacto, disco de video digital, disco Blu-ray, u otro medio legible por ordenador. En algunos ejemplos, un servidor de red (que no se muestra) puede recibir datos de video codificados procedentes del dispositivo fuente 12 y puede proporcionar los datos de video codificados al dispositivo destino 14, por ejemplo, a través de la transmisión de red. De manera similar, un dispositivo informático de una instalación de producción de medio, tal como una instalación de estampado de disco, puede recibir datos de video codificados procedentes del dispositivo fuente 12 y puede producir un disco que contiene los datos de video codificados. Por lo tanto, se puede entender 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 destino 14 recibe información desde el medio 16 legible por ordenador. La información del medio 16 legible por ordenador puede incluir información de sintaxis definida por el codificador 20 de video, que también es utilizada por el decodificador 30 de video, que incluye elementos de sintaxis que describen características y/o procesamiento de bloques y otras unidades codificadas, por ejemplo, grupo de imágenes (GOP). El dispositivo 32 de visualización muestra los datos de video decodificados a un usuario, y puede comprender cualquiera de una variedad de dispositivos de visualización tal como un tubo de rayos catódicos (CRT), una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodo de emisión de luz orgánica (OLED), u otro tipo de dispositivo de visualización.

El codificador 20 de video y el decodificador 30 de video pueden operar de acuerdo con una norma de codificación de video, tal como la norma de Codificación de Video de Alta Eficiencia (HEVC) actualmente bajo desarrollo, y se pueden adaptar al Modelo de Prueba HEVC (HM). Alternativamente, el codificador 20 de video y el decodificador 30 de video pueden operar de acuerdo con otras normas de propiedad o industriales, tal como la norma del Sector de Estandarización de Telecomunicaciones de la Unión de Telecomunicaciones Internacional (ITU-T) H.264, alternativamente denominado como Grupo de Expertos de Imágenes en

Movimiento (MPEG)-4, Parte 10, Codificación de Video Avanzada (AVC), H.265/HEVC, o extensiones de dichas normas. Las técnicas de esta descripción, no obstante, no se limitan a ninguna norma de codificación particular. Otros ejemplos de normas de codificación de videos incluyen MPEG-2 e ITU-T H.263. Aunque no se muestra en la figura 1, en algunos aspectos, el codificador 20 de video y el decodificador 30 de video pueden estar integrados con un codificador y decodificador de audio, y pueden incluir unidades de multiplexor-desmultiplexor apropiados (MUX-DEMUX), u otro hardware y software, para manejar la codificación tanto de audio como de video en una corriente de datos común o corrientes de datos separadas. Si es aplicable, se pueden adaptar unidades MUX-DEMUX al protocolo de multiplexor ITU H.223, u otros protocolos tal como el protocolo de datagrama de usuario (UDP).

El codificador 20 de video y el decodificador 30 de video pueden ser implementados como cualquiera de una variedad de circuitos de codificador adecuados, tal como uno o más microprocesadores, procesadores de señal digital (DSP), circuitos integrados de aplicación específica (ASIC), agrupaciones de puerta programables en campo (FPGA), lógica discreta, software, hardware, firmware o cualesquiera combinaciones de los mismos. Cuando las técnicas son implementadas parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio legible por ordenador no transitorio, adecuado y puede realizar las instrucciones en hardware utilizando uno o más procesadores para realizar las técnicas de esta descripción. Cada uno del codificador 20 de video y del decodificador 30 de video puede ser incluido en uno o más codificadores o decodificadores, cualquiera de los cuales puede estar integrado como parte de un codificador/decodificador (CÓDEC) combinados en un dispositivo respectivo. Un dispositivo que incluye el codificador 20 de video y/o el decodificador 30 de video puede comprender 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 un codificador 20 de video que puede implementar técnicas de codificación de video. El codificador 20 de video puede ejecutar intra- e inter-codificación de bloques de video dentro de los segmentos de video. La intra-codificación se basa en la predicción espacial para reducir o eliminar la redundancia espacial en video dentro de un fotograma de video o imagen determinado. La inter-codificación se basa en predicción temporal para reducir o eliminar la redundancia temporal en video dentro de fotogramas o imágenes adyacentes de una secuencia de video. El intra-modo (modo I) se puede referir a cualquiera de los diversos modos de codificación basados espaciales. Los inter-modos, tales como predicción uni-direccional (también conocida como, uni-predicción) (modo P) o bi-predicción (también conocida como, bi-predicción) (modo B), se pueden referir a cualquiera de varios modos de codificación basados temporales.

Como se muestra en la figura 2, el codificador 20 de video recibe un bloque de video actual dentro de un fotograma de video que se ha de codificar. En el ejemplo de la figura 2, el codificador 20 de video incluye la unidad 40 de selección de modo, la memoria 64 de fotogramas de referencia, la sumadora 50, la unidad 52 de procesamiento de transformada, la unidad 54 de cuantificación, y unidad 56 de codificación entrópica. La unidad 40 de selección de modo, a su vez, incluye la unidad 44 de compensación de movimiento, la unidad 42 de estimación de campo de movimiento, unidad 46 de intra-predicción (también conocida como, intra-predicción), y la unidad 48 de división en particiones. Para reconstrucción de bloques de video, el codificador 20 de video también incluye la unidad 58 de cuantificación inversa, la unidad 60 de transformada inversa, y la sumadora 62. Un filtro de desbloqueo (no mostrado en la figura 2) también se puede incluir para filtrar límites de bloque para eliminar los artefactos de bloqueo del video reconstruido. Si se desea, el filtro de desbloqueo típicamente filtraría la salida de la sumadora 62. También se pueden utilizar filtros adicionales (en-bucle o post-bucle) además del filtro de desbloqueo. Dichos filtros no se muestran por brevedad, pero si se desea, pueden filtrar la salida de la sumadora 50 (como un filtro en-bucle).

Durante el proceso de codificación, el codificador 20 de video recibe un fotograma o segmento de video que se ha de codificar. El fotograma o segmento puede ser dividido en múltiples bloques de video. La unidad 42 de estimación de movimiento y la unidad 44 de compensación de movimiento ejecutan codificación inter-predictiva del bloque de video recibido con relación a uno o más bloques en uno o más fotogramas de referencia para proporcionar predicción temporal. La unidad 46 de intra-predicción alternativamente puede realizar codificación intra-predictiva del bloque de video recibido con relación a uno o más bloques contiguos en el mismo fotograma o segmento que el bloque que se ha de codificar para proporcionar predicción espacial. El codificador 20 de video puede realizar múltiples pasadas de codificación, por ejemplo, para seleccionar un modo de codificación apropiado para cada bloque de datos de video.

Además, la unidad 48 de partición puede dividir en particiones bloques de datos de video en sub-bloques, basándose en la evaluación de esquemas de partición previos en pasadas de codificación previas. Por ejemplo, la unidad 48 de partición inicialmente puede dividir en particiones un fotograma o segmento en unidades de codificación más grandes (LCU), y puede dividir en particiones cada una de las LCU en unidades de sub-codificación (sub-CU) basándose en análisis de distorsión de tasa (por ejemplo, optimización de distorsión de tasa). La unidad 40 de selección de modo además puede producir una estructura de datos de árbol cuaternario indicativa de la partición de una LCU en sub-CU. Las CU de nodo hoja del árbol cuaternario pueden incluir una o más unidades de predicción (PU) y una o más unidades de transformada (TU).

La presente descripción utiliza 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 otras normas (por ejemplo, macrobloques y sub-bloques 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 puede variar desde 8×8 píxeles hasta el tamaño del bloque de árbol con un máximo de 64×64 píxeles o mayor. Cada CU puede contener una o más PU y una o más TU. Datos de sintaxis asociados con una CU pueden describir, por ejemplo, la partición de la CU en una o más PU. Los modos de partición pueden diferir entre sí si la CU es omitida o codificada en modo directo, codificada en modo de intra-predicción, o codificada en modo de inter-predicción (también conocida como, inter-predicción). Las PU pueden ser divididas en particiones para no tener forma cuadrada. Datos de sintaxis asociados con una CU también pueden describir, por ejemplo, la partición de la CU en una o más TU según un árbol cuaternario. Una TU puede ser de forma cuadrada o no cuadrada (por ejemplo, rectangular).

La unidad 40 de selección de modo puede seleccionar uno de los modos de codificación, intra- o inter-, por ejemplo, basándose en los resultados de error, y proporciona el bloque intra- o intercodificado resultante a la sumadora 50 para generar datos de bloque residuales y a la sumadora 62 para reconstruir el bloque codificado para usar como un fotograma de referencia. La unidad 40 de selección de modo también proporciona elementos de sintaxis, tal como vectores de movimiento, indicadores de intra-modo, información de partición, y otra información de sintaxis de este tipo, a la unidad 56 de codificación entrópica.

La unidad 42 de estimación de movimiento y la unidad 44 de compensación de movimiento pueden estar muy integradas, pero son ilustradas separadamente para propósitos conceptuales. La estimación de movimiento, realizada por la unidad 42 de estimación de movimiento, es el proceso de generar vectores de movimiento, los cuales estiman el movimiento para los bloques de video. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de una PU de un bloque de video dentro de un fotograma de video o imagen actual con relación a un bloque predictivo dentro de un fotograma de referencia (u otra unidad codificada) con relación al bloque actual que está siendo codificado dentro del fotograma actual (u otra unidad codificada). Un bloque predictivo es un bloque que se empareja muy estrechamente con el bloque que ha de ser codificado, en términos de diferencia de pixel, lo cual puede ser determinado por la suma de diferencia absoluta (SAD), suma de diferencia cuadrada (SSD) u otras métricas de diferencia. En algunos ejemplos, el codificador 20 de video puede calcular valores para posiciones de pixel de subentero de imágenes de referencia almacenadas en la memoria 64 de fotogramas de referencia. Por ejemplo, el codificador 20 de video puede interpolar valores de posiciones de pixel de un cuarto, posiciones de pixel de un octavo u otras posiciones de pixel fraccionarias de la imagen de referencia. Por lo tanto, la unidad 42 de estimación de movimiento puede realizar una búsqueda de movimiento con relación a las posiciones de pixel completo y posiciones de pixel fraccionarias y puede emitir un vector de movimiento con precisión de pixel fraccionaria.

La unidad 42 de estimación de movimiento calcula un vector de movimiento para una PU de un bloque de video en un segmento intercodificado comparando la posición de la PU con la posición de un bloque predictivo de una imagen de referencia. La imagen de referencia puede ser seleccionada de entre 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 entrópica y a la unidad 44 de compensación de movimiento.

La compensación de movimiento, realizada por la unidad 44 de compensación de movimiento, puede involucrar la extracción o generación del bloque predictivo con base en el vector de movimiento determinado por la unidad 42 de estimación de movimiento. De nuevo, la unidad 42 de estimación de movimiento y la unidad 44 de compensación de movimiento pueden estar funcionalmente integradas, en algunos ejemplos. Después de recibir el vector de movimiento para la PU del bloque de video actual, la unidad 44 de compensación de movimiento puede localizar el bloque predictivo al cual apunta el vector de movimiento en una de las listas de imágenes de referencia. La sumadora 50 forma un bloque de video residual sustrayendo valores de pixel del bloque predictivo a partir de los valores de pixel del bloque de video actual que se está codificando, formando valores de diferencia de pixel, como se analiza a continuación. En general, la unidad 42 de estimación de movimiento realiza estimación de movimiento con relación a componentes de luma, y la unidad 44 de compensación de movimiento utiliza vectores de movimiento calculados basándose en los componentes de luma tanto para componentes de croma como para componentes de luma. La unidad 40 de selección de modo también puede generar elementos de sintaxis asociados con los bloques de video y el segmento de video para usar por el decodificador 30 de video en la decodificación de los bloques de video del segmento de video.

La unidad 46 de intra-predicción puede intra-predecir un bloque actual, como una alternativa a la inter-predicción ejecutada por la unidad 42 de estimación de movimiento y la unidad 44 de compensación de movimiento, como se describió antes. En particular, la unidad 46 de intra-predicción puede determinar un modo de intra-predicción a utilizar para codificar un bloque actual. En algunos ejemplos, la unidad 46 de intra-predicción puede codificar un bloque actual utilizando varios modos de intra-predicción, por ejemplo, durante pasadas de codificación separadas, y la unidad 46 de intra-predicción (o unidad 40 de selección de modo, en

algunos ejemplos) puede seleccionar un modo de intra-predicción apropiado a utilizar de entre los modos probados.

5 Por ejemplo, la unidad 46 de intra-predicción puede calcular valores de distorsión de tasa utilizando un análisis de distorsión de tasa para los diversos modos de intra-predicción probados, y seleccionar el modo de  
 10 intra-predicción 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 no codificado original que fue codificado para producir el bloque codificado, así como una tasa de bits (es decir, un número de bits) utilizada para producir el bloque codificado. La unidad 46 de intra-predicción puede calcular relaciones de las distorsiones y tasas para los diversos bloques codificados para determinar qué modo de intra-predicción muestra el mejor valor de distorsión de tasa para el bloque.

15 Además, la unidad 46 de intra-predicción puede ser configurada para codificar bloques de profundidad de un mapa de profundidad utilizando un modo de modelado de profundidad (DMM). La unidad 40 de selección de modo puede determinar si un modo DMM disponible produce mejores resultados de codificación que un modo de intra-predicción y los otros modos DMM, por ejemplo, utilizando optimización de distorsión de tasa (RDO). Datos para una imagen de textura correspondientes a un mapa de profundidad pueden ser almacenados en la memoria 64 de fotogramas de referencia. La unidad 42 de estimación de movimiento y la  
 20 unidad 44 de compensación de movimiento también se pueden configurar para inter-prededir bloques de profundidad de un mapa de profundidad.

25 Después de seleccionar un modo de intra-predicción para un bloque (por ejemplo, un modo de intra-predicción convencional o uno de los modos DMM), la unidad 46 de intra-predicción puede proporcionar información indicativa del modo de intra-predicción seleccionado para el bloque a la unidad 56 de codificación entrópica. La unidad 56 de codificación entrópica puede codificar la información indicando el modo de intra-predicción seleccionado. El codificador 20 de video puede incluir, en los datos de configuración de corriente de bits transmitidos, que pueden incluir una pluralidad de tablas de índice de modo de intra-predicción y una pluralidad de tablas de índice de modo de intra-predicción modificado (también denominadas como tablas de correspondencia o mapeo de palabra código), definiciones de contextos de codificación para varios bloques, e indicaciones de un modo de intra-predicción más probable, una tabla de índice de modo de intra-predicción, y una tabla de índice de modo de intra-predicción modificado a utilizar para cada uno de los contextos.

30 El codificador 20 de video forma un bloque de video residual sustrayendo los datos de predicción procedentes de la unidad 40 de selección de modo a partir del bloque de video original que está siendo codificado. La sumadora 50 representa el componente o componentes que realizan esta operación de sustracción.

35 La unidad 52 de procesamiento de transformada aplica una transformada, tal como una transformada de coseno discreto (DCT) o una transformada conceptualmente similar, al bloque residual, produciendo un bloque de video que comprende valores de coeficiente de transformada residuales. La unidad 52 de procesamiento de transformada puede realizar otras transformadas que son conceptualmente similares a la DCT. También se podrían utilizar transformadas de ondícula, transformadas de número entero, transformadas de sub-banda u otros tipos de transformadas.

40 La unidad 52 de procesamiento de transformada aplica la transformada al bloque residual, produciendo un bloque de coeficientes de transformada residuales. La transformada puede convertir la información residual procedente de un dominio de valor de pixel a un dominio de transformada, tal como un dominio de frecuencia. La unidad 52 de procesamiento de transformada puede enviar los coeficientes de transformada resultantes a la unidad 54 de cuantificación. La unidad 54 de cuantificación cuantifica los coeficientes de transformada para reducir adicionalmente la tasa de bits. El proceso de cuantificación puede reducir la profundidad de bit asociada con algunos o todos los coeficientes. El grado de cuantificación puede ser modificado ajustando un parámetro de cuantificación. En algunos ejemplos, la unidad 54 de cuantificación entonces puede realizar un escaneo de la matriz incluyendo los coeficientes de transformada cuantificados. De manera alternativa, la unidad 56 de codificación entrópica puede realizar el escaneo.

50 Después de la cuantificación, la unidad 56 de codificación entrópica codifica por entropía los coeficientes de transformada cuantificados. Por ejemplo, la unidad 56 de codificación entrópica puede realizar codificación de longitud variable adaptativa al contexto (CAVLC), codificación aritmética binaria adaptativa al contexto (CABAC), codificación aritmética binaria adaptativa al contexto basada en sintaxis (SBAC), codificación entrópica de división de intervalo de probabilidad (PIPE) u otra técnica de codificación entrópica. En el caso de codificación entrópica basada en contexto, el contexto se puede basar en bloques contiguos. Después de  
 55 la codificación entrópica por la unidad 56 de codificación entrópica, la corriente de bits codificada puede ser transmitida a otro dispositivo (por ejemplo, decodificador 30 de video) o archivada para 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 pixel, por

ejemplo, para posterior uso como un bloque de referencia. La unidad 44 de compensación de movimiento puede calcular 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 puede aplicar uno o más filtros de interpolación al bloque residual reconstruido para calcular valores de pixel de subentero para usar en estimación de movimiento. La sumadora 62 suma el bloque residual reconstruido al bloque de predicción de movimiento compensado producido por la unidad 44 de compensación de movimiento para producir un bloque de video reconstruido para almacenamiento en la memoria 64 de fotogramas de referencia. El bloque de video reconstruido puede ser utilizado por la unidad 42 de estimación de movimiento y la unidad 44 de compensación de movimiento como un bloque de referencia para inter-codificar un bloque en un fotograma de video subsiguiente.

La figura 3 es un diagrama de bloques que ilustra un ejemplo de un decodificador 30 de video que puede implementar técnicas de codificación de video. En el ejemplo de la figura 3, el decodificador 30 de video incluye una unidad 70 de decodificación entrópica, una unidad 72 de compensación de movimiento, una unidad 74 de intra-predicción, una unidad 76 de cuantificación inversa, una unidad 78 de transformación inversa, una memoria 82 de fotogramas de referencia, y una sumadora 80. El decodificador 30 de video, en algunos ejemplos, puede realizar una pasada de decodificación generalmente recíproca a la pasada de codificación descrita con respecto al codificador 20 de video (figura 2). La unidad 72 de compensación de movimiento puede generar datos de predicción basándose en vectores de movimiento recibidos desde la unidad 70 de decodificación entrópica, mientras que la unidad 74 de intra-predicción puede generar datos de predicción basándose en los indicadores de modo de intra-predicción recibidos desde la unidad 70 de decodificación entrópica.

Durante el proceso de decodificación, el decodificador 30 de video recibe una corriente de bits de video codificada que representa bloques de video de un segmento de video codificado y elementos de sintaxis asociados procedentes del codificador 20 de video. La unidad 70 de decodificación entrópica del decodificador 30 de video decodifica por entropía la corriente de bits para generar coeficientes cuantificados, vectores de movimiento o indicadores de modo de intra-predicción, y otros elementos de sintaxis. La unidad 70 de decodificación entrópica reenvía los vectores de movimiento y otros elementos de sintaxis a la unidad 72 de compensación de movimiento. El decodificador 30 de video puede recibir los elementos de sintaxis al nivel de segmento de video y/o al nivel de bloque de video.

Cuando el segmento de video es codificado como un segmento intra-codificado (I), la unidad 74 de intra-predicción puede generar datos de predicción para un bloque de video del segmento de video actual basándose en un modo de intra-predicción señalado y datos de bloques previamente decodificados del fotograma o imagen actual. Cuando el fotograma de video es codificado como un segmento intercodificado (por ejemplo, B, P, o GPB), la unidad 72 de compensación de movimiento produce bloques predictivos para un bloque de video del segmento de video actual basándose en los vectores de movimiento y otros elementos de sintaxis recibidos procedentes de la unidad 70 de decodificación entrópica. Los bloques predictivos pueden ser producidos a partir de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. El decodificador 30 de video puede construir las listas de fotogramas de referencia, Lista 0 y Lista 1, utilizando técnicas de construcción por defecto basándose en 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 video del segmento de video actual analizando los vectores de movimiento y otros elementos de sintaxis, y utiliza la información de predicción para producir los bloques predictivos para el bloque de video actual que se está decodificando. Por ejemplo, la unidad 72 de compensación de movimiento utiliza algunos de los elementos de sintaxis recibidos para determinar un modo de predicción (por ejemplo, intra- o inter-predicción) utilizado para codificar los bloques de video del segmento de video, un tipo de segmento de inter-predicción (por ejemplo, segmento B, segmento P, o segmento GPB), información de construcción para una o más de las listas de imágenes de referencia para el segmento, vectores de movimiento para cada bloque de video intercodificado del segmento, estatus de inter-predicción para cada bloque de video intercodificado del segmento, y otra información para decodificar los bloques de video en el segmento de video actual.

La unidad 72 de compensación de movimiento también puede realizar interpolación basándose en filtros de interpolación. La unidad 72 de compensación de movimiento puede utilizar filtros de interpolación tal como son utilizados por el codificador 20 de video durante la codificación de los bloques de video para calcular valores interpolados para píxeles de subentero de bloques de referencia. En este caso, la unidad 72 de compensación de movimiento puede determinar los filtros de interpolación utilizados por el codificador 20 de video a partir de los elementos de sintaxis recibidos y utilizar los filtros de interpolación para producir bloques predictivos.

Datos para una imagen de textura correspondientes a un mapa de profundidad pueden ser almacenados en la memoria 82 de fotogramas de referencia. La unidad 72 de compensación de movimiento también puede ser configurada para inter-prededir bloques de profundidad de un mapa de profundidad.

La compresión de imagen y video ha experimentado un rápido crecimiento, conduciendo a varias normas de codificación. Dichas normas de codificación de video incluyen el Sector de Estandarización de Telecomunicaciones (ITU-T) H.261 de la Unión de Telecomunicaciones Internacional (ITU), Grupo de Expertos de Imágenes en Movimiento ISO/IEC (MPEG)-1 Parte 2, ITU-T H.262 o ISO/IEC MPEG-2 Parte 2, 5 ITU-T H.263, ISO/IEC MPEG-4 Parte 2, Codificación de Video Avanzada (AVC), también conocida como ITU-T H.264 o ISO/IEC MPEG-4 Parte 10, y Codificación de Video de Alta Eficiencia (HEVC), también conocida como ITU-T H.265 o MPEG-H Parte 2. La AVC incluye extensiones tales como Codificación de Video Escalable (SVC), Codificación de Video Multivisión (MVC) y Codificación de Video Multivisión más Profundidad (MVC+D), y 3D AVC (3D-AVC). HEVC incluye extensiones tales como HEVC Escalable (SHVC), 10 HEVC Multivisión (MV-HEVC), y 3D HEVC (3D-HEVC).

La Codificación de Video Versátil (VVC) es una nueva norma de codificación de video bajo desarrollo por el equipo de expertos en video conjuntos (JVET) de ITU-T e ISO/IEC. En el momento de escribir esto, el Borrador de Trabajo (WD) de VVC más reciente está incluido en JVET-K1001-v1. El documento JVET-JVET-K0325-v3 incluye una actualización a la sintaxis de alto nivel de VVC.

15 La presente descripción describe técnicas destinadas a abordar el sub-desarrollo de la norma VVC. Sin embargo, las técnicas también se pueden aplicar a otras especificaciones de códec de video/medios.

Técnicas de compresión de video realizan predicción espacial (intra-imagen) y/o predicción temporal (inter-imagen) para reducir o remover la redundancia inherente en secuencias de video. Para codificación de video basada en bloque, un segmento de video (por ejemplo, una imagen de video o una porción de una imagen de video) puede ser dividida en particiones en bloques de video, los cuales también pueden ser denominados como bloques de árbol, bloques de árbol de codificación (CTB), unidades de árbol de codificación (CTU), 20 unidades de codificación (CU), y/o nodos de codificación. Los bloques de video en un segmento intra-codificado (I) de una imagen son codificados utilizando predicción espacial con respecto a muestras de referencia en bloques contiguos en la misma imagen. Bloques de video en un segmento intercodificado (P o B) de una imagen pueden utilizar predicción espacial con respecto a muestras de referencia en bloques contiguos en la misma imagen o predicción temporal con respecto a muestras de referencia en otras imágenes de referencia. Las imágenes pueden ser denominadas como fotogramas, y las imágenes de referencia pueden ser denominadas como 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 pixel entre el bloque original que se ha de codificar y el bloque predictivo. Un bloque intercodificado es codificado 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 intra-codificado es codificado según un modo de intra-codificación y los datos residuales. Para compresión adicional, los datos residuales pueden ser transformados desde el dominio de pixel a un dominio de transformada, dando como resultado coeficientes de transformada residuales, que luego pueden ser cuantificados. Los coeficientes de transformada cuantificados, dispuestos inicialmente en una disposición bidimensional, pueden ser escaneados para producir un vector unidimensional de coeficientes de transformada, y se puede aplicar codificación entrópica para lograr incluso más compresión.

40 En una especificación de video códec, las imágenes son identificadas para múltiples propósitos, incluyendo para uso como una imagen de referencia en inter-predicción, para emitir imágenes desde la memoria tampón de imágenes decodificadas (DPB), para escalar vectores de movimiento, para predicción ponderada, etc. En AVC y HEVC, las imágenes pueden ser identificadas por el recuento de orden de imágenes (POC). En AVC y HEVC, las imágenes en la DPB pueden ser marcadas como "utilizadas para referencia a corto plazo," 45 "utilizadas para referencia a largo plazo," o "no utilizada para referencia." Una vez que una imagen ha sido marcada "no utilizada para referencia" ya no puede ser utilizada para predicción, y cuando ya no es necesaria para emisión, puede ser eliminada de la DPB.

En AVC, hay dos tipos de imágenes de referencia, a largo plazo y a corto plazo. Una imagen de referencia puede ser marcada como "no utilizada para referencia" cuando ya no resulta necesaria para referencia de predicción. La conversión entre estos tres estatus (corto plazo, largo plazo y no utilizada para referencia) es controlada por el proceso de marcación de imágenes de referencia decodificadas. Hay dos mecanismos de marcación alternativos de imágenes de referencia decodificadas, el proceso implícito de ventana deslizante y el proceso explícito de operación de control de gestión de memoria (MMCO). 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 determinado (max\_num\_ref\_frames en el SPS). Las imágenes de referencia a corto plazo son almacenadas en una manera de primera en entrar, primera en salir de modo que las imágenes a corto plazo más recientemente decodificadas se mantienen en la DPB.

60 El proceso explícito de MMCO puede incluir múltiples comandos de MMCO. Un comando de MMCO puede marcar una o más imágenes de referencia a corto plazo o a largo plazo como "no utilizada para referencia,"

marcar todas las imágenes como “no utilizada para referencia,” o marcar la imagen de referencia actual o una imagen de referencia a corto plazo existente como a largo plazo, y asignar un índice de imagen a largo plazo a esa imagen de referencia a largo plazo.

5 En AVC, las operaciones de marcación de imágenes de referencia así como los procesos para emitir y eliminar imágenes de la DPB son realizados después que una imagen ha sido decodificada.

10 HEVC introduce un enfoque diferente para la gestión de imágenes de referencia, denominado como conjunto de imágenes de referencia (RPS). La diferencia más fundamental con el concepto RPS comparado con MMCO/ventana deslizante de AVC es que para cada segmento particular se proporciona un conjunto completo de las imágenes de referencia que son utilizadas por la imagen actual o cualquier imagen subsiguiente. Así, se señala un conjunto completo de todas las imágenes que deben ser mantenidas en la DPB para usar por la imagen actual o futura. Esto es diferente del esquema AVC donde únicamente se señalan cambios relativos a la DPB. Con el concepto RPS, no se necesita información de imágenes previas en el orden de decodificación para mantener el estatus correcto de imágenes de referencia en la DPB.

15 El orden de las operaciones de decodificación de imágenes y DPB en HEVC se cambia en comparación con AVC para explotar las ventajas de RPS y mejorar la resiliencia de error. En AVC las operaciones de marcación de imágenes y memoria tampón, tanto la salida como la eliminación de imágenes decodificadas procedentes de la DPB generalmente son aplicadas después que una imagen actual ha sido decodificada. En HEVC, el RPS primero es decodificado procedente de una cabecera de segmento de la imagen actual, luego se aplican generalmente las operaciones de marcación de imágenes y memoria tampón antes de decodificar la imagen actual.

20 Cada cabecera de segmento en HEVC debe incluir parámetros para señalar el RPS para la imagen que contiene los segmentos. La única excepción es que ningún RPS es señalado para segmentos IDR, en vez de ello se infiere que el RPS está vacío. Para segmentos I que no pertenecen a una imagen IDR, se puede prever un RPS, incluso si pertenecen a una imagen I debido a que puede haber imágenes después de la imagen I en el orden de decodificación que utilizan inter-predicción a partir de imágenes que precedieron a la imagen I en el orden de decodificación. El número de imágenes en un RPS no excederá el límite de tamaño de DPB conforme a lo 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. Las cabeceras de segmentos contienen una palabra código de longitud fija, `pic_order_cnt_lsb`, que representa los bits menos significativos del valor POC completo, también conocido como el POC LSB. La longitud de la palabra código es señalada en el SPS y puede ser de entre 4 y 16 bits. El concepto de RPS utiliza POC para identificar imágenes de referencia. Además de su propio valor POC, cada cabecera de segmento contiene o hereda directamente del SPS una representación codificada de los valores POC (o los LSB) de cada imagen en el RPS.

35 El RPS para cada imagen incluye cinco listas diferentes de imágenes de referencia, también referidas a los cinco subconjuntos RPS. `RefPicSetStCurrBefore` incluye todas las imágenes de referencia a corto plazo que son anteriores a la imagen actual tanto en el orden de decodificación como en el orden de emisión, y que pueden ser utilizadas en inter-predicción de la imagen actual. `RefPicSetStCurrAfter` incluye todas las imágenes de referencia a corto plazo que son anteriores a la imagen actual en el orden de decodificación, que siguen a la imagen actual en el orden de emisión, y que pueden ser utilizadas en inter-predicción de la imagen actual. `RefPicSetStFoll` incluye todas las imágenes de referencia a corto plazo que pueden ser utilizadas en inter-predicción de una o más de las imágenes que siguen a la imagen actual en el orden de decodificación, y que no son utilizadas en la inter-predicción de la imagen actual. `RefPicSetLtCurr` incluye todas las imágenes de referencia a largo plazo que pueden ser utilizadas en inter-predicción de la imagen actual. `RefPicSetLtFoll` incluye todas las imágenes de referencia a largo plazo que pueden ser utilizadas en inter-predicción de una o más de las imágenes que siguen a la imagen actual en orden de decodificación, y que no son utilizadas en la inter-predicción de la imagen actual.

50 El RPS es señalado utilizando hasta tres bucles que se repiten sobre diferentes tipos de imágenes de referencia; imágenes de referencia a corto plazo con valor POC inferior a la imagen actual, imágenes de referencia a corto plazo con valor POC superior a la imagen actual, e imágenes de referencia a largo plazo. Además, un indicador (`used_by_curr_pic_X_flag`) es enviado para cada imagen de referencia indicando 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 en el subconjunto 402 denominada como `RefPicSetStCurrBefore` debido a que la imagen está antes en el orden de salida y es utilizada por B14. P12 es la imagen en el subconjunto

402 denominada como RefPicSetStCurrAfter debido a que la imagen está después en el orden de salida y es utilizada por B14. P13 es la imagen en el subconjunto 402 denominada como RefPicSetStFoll debido a que la imagen es una imagen de referencia a corto plazo no utilizada por B14 (pero debe mantenerse en la DPB debido a que es utilizada por B15). P4 es la imagen en el subconjunto 402 denominada como RefPicSetLtCurr debido a que la imagen es una imagen de referencia a largo plazo utilizada por B14. I0 es la imagen en el subconjunto 402 denominada como RefPicSetLtFoll dado que la imagen es una imagen de referencia a largo plazo no utilizada por la imagen actual (pero debe ser mantenida en la DPB dado que es utilizada por B15).

La parte a corto plazo del RPS 400 puede ser incluida directamente en la cabecera de segmento. Alternativamente, la cabecera de segmento puede contener solamente 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 puede ser señalizada utilizando cualquiera de dos esquemas diferentes; Inter RPS, como se describe a continuación, o Intra RPS, como se describe aquí. Cuando se utiliza Intra RPS, num\_negative\_pics y num\_positive\_pics son señalizadas representando la longitud de dos listas diferentes de imágenes de referencia. Estas listas contienen las imágenes de referencia con diferencia POC negativa y diferencia POC positiva en comparación con la imagen actual, respectivamente. Cada elemento en estas listas es codificado con un código de longitud variable que representa la diferencia en el valor POC con relación al elemento previo en la lista menos uno. Para la primera imagen en cada lista, la señalización es relativa al valor POC de la imagen actual menos uno.

Cuando se codifican 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 es denominado como Inter RPS. No hay problemas de robustez de error asociados con este método debido a que todos los RPS del conjunto de parámetros de secuencia están en la misma unidad de capa de abstracción de red (NAL). La sintaxis Inter RPS aprovecha el hecho de que el RPS de la imagen actual puede ser predicho 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, o bien imágenes de referencia de la imagen previa, o bien la propia imagen previamente decodificada. Solamente es necesario indicar cuáles de estas imágenes deben ser las imágenes de referencia y ser utilizadas para la predicción de la imagen actual. Por lo tanto, la sintaxis comprende lo siguiente: un índice que apunta al RPS para utilizarse como un predictor, un delta POC que ha de ser añadido 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 solamente son utilizadas para la predicción de imágenes futuras. En una realización, delta POC se refiere a la diferencia en el valor POC entre una imagen de referencia actual y otra imagen de referencia (por ejemplo, previa).

Los codificadores que les gustaría aprovechar el uso de imágenes de referencia a largo plazo deben establecer el elemento de sintaxis SPS long\_term\_ref\_pics\_present\_flag a uno. Las imágenes de referencia a largo plazo entonces pueden ser señalizadas en la cabecera de segmento por 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 fue señalizada para una imagen a largo plazo particular. También es posible señalar un conjunto de imágenes a largo plazo en el SPS como una lista de valores POC LSB. El POC LSB para una imagen a largo plazo entonces puede ser señalizado en la cabecera de segmento como un índice para esta lista.}

El elemento de sintaxis delta\_poc\_msb\_cycle\_lt\_minus1 adicionalmente puede ser señalizado para permitir el cálculo de la distancia POC completa de una imagen de referencia a largo plazo con relación a la imagen actual. Se requiere que la palabra código delta\_poc\_msb\_cycle\_lt\_minus1 sea señalizada para cada imagen de referencia a largo plazo que tiene el mismo valor POC LSB que cualquier otra imagen de referencia en el RPS.

Para marcación de imagen de referencia en HEVC, normalmente habrá un número de imágenes presentes en la DPB antes de la decodificación de imágenes. Algunas de ellas pueden estar disponibles para predicción y así ser marcadas como "utilizadas para referencia." Otras pueden no estar disponibles para predicción pero pueden estar a la espera de emisión, así marcadas como "no utilizadas para referencia". Cuando la cabecera de segmento ha sido analizada, se lleva a cabo un proceso de marcación de imagen antes que se decodifiquen los datos de segmento. Las imágenes que están presentes en la DPB y que están marcadas como "utilizadas para referencia" pero no están incluidas en el RPS son marcadas como "no utilizadas para referencia." Las imágenes que no están presentes en la DPB pero que están incluidas en el conjunto de imágenes de referencia son ignoradas si el used\_by\_curr\_pic\_X\_flag utilizado es igual a cero. Sin embargo, si en vez de ello el used\_by\_curr\_pic\_X\_flag utilizado es igual a uno, esta imagen de referencia fue destinada a ser utilizada para predicción en la imagen actual pero se ha perdido. Luego, se infiere una pérdida de imagen no intencionada y el decodificador debiera emprender las acciones apropiadas.

Después de decodificar la imagen actual, esta es marcada como "utilizada para referencia a corto plazo."

A continuación se analiza la construcción de la lista de imágenes de referencia en HEVC. En HEVC, el término inter-predicción es utilizado para denotar predicción derivada de elementos de datos (por ejemplo, valores de muestra o vectores de movimiento) de imágenes de referencia distintos de la imagen decodificada actual. Al igual que en AVC, una imagen puede ser predicha a partir de múltiples imágenes de referencia. Las imágenes de referencia que son utilizadas para inter-predicción son organizadas en una o más listas de imágenes de referencia. El índice de referencia identifica cuál de las imágenes de referencia en la lista debe ser utilizada para crear la señal de predicción.

Una sola lista de imágenes de referencia, Lista 0, es utilizada para un segmento P y dos listas de imágenes de referencia, Lista 0 y Lista 1 son utilizadas para segmentos B. De modo 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 segmentos P (para los cuales se utiliza el orden de decodificación) y segmentos B (para los cuales 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 Lista 1 por defecto (si el segmento es un segmento B) basándose en tres subconjuntos RPS: RefPicSetStCurrBefore, RefPicSetStCurrAfter, y RefPicSetLtCurr. Imágenes a corto plazo con orden de emisión anterior (posterior) son insertadas en primer lugar en la Lista 0 (Lista 1) en orden ascendente de distancia POC a la imagen actual, luego las imágenes a corto plazo con orden de emisión posterior (anterior) son insertadas en la Lista 0 (Lista 1) en orden ascendente de distancia POC a la imagen actual, y finalmente las imágenes a largo plazo son insertadas al final. En términos de RPS, para la Lista 0, las entradas en RefPicSetStCurrBefore son insertadas en la lista inicial, seguidas por las entradas en RefPicSetStCurrAfter. Posteriormente, las entradas en RefPicSetLtCurr, en caso de estar disponibles, son añadidas.

En HEVC, se repite el proceso anterior (imágenes de referencia que ya han sido añadidas a la lista de imágenes de referencia son añadidas de nuevo) cuando el número de entradas en una lista es más pequeño que el número objetivo de imágenes de referencia activas (señalizadas en el conjunto de parámetros de imagen o cabecera de segmento). Cuando el número de entradas es más grande que el número objetivo, la lista es truncada.

Después que se ha inicializado una lista de imágenes de referencia, puede ser modificada de manera que las imágenes de referencia para la imagen actual pueden ser dispuestas en cualquier orden, incluyendo el caso donde una imagen de referencia particular puede aparecer en más de una posición en la lista, basándose en los comandos de modificación de lista de imágenes de referencia. Cuando el indicador que indica la presencia de modificaciones de lista se establece en uno, se señala 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 es identificada en el comando por el índice para la lista de imágenes de referencia para la imagen actual derivada de la señalización RPS. Esto es diferente de la modificación de lista de imágenes de referencia en H.264/AVC, en donde una imagen es identificada ya sea por el número de imagen (derivado del elemento de sintaxis frame\_num) o el índice de imágenes de referencia a largo plazo, y es posible que se necesiten menos comandos por ejemplo, para intercambiar las primeras dos entradas de una lista inicial o insertar una entrada al comienzo de la lista inicial y desplazar los otros.

Una lista de imágenes de referencia no tiene permitido incluir ninguna imagen de referencia con el TemporalId mayor que la imagen actual. Una corriente de bits HEVC podría consistir de varias sub-capas temporales. Cada unidad NAL pertenece a una sub-capa específica como es indicado por el TemporalId (igual a temporal\_id\_plus1 - 1).

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

Desafortunadamente la gestión de imágenes de referencia basada en Listas de Imágenes de Referencia (RPL) tiene inconvenientes. Por ejemplo, algunos esquemas de gestión de imágenes de referencia basados en RPL no son optimizados en términos de la señalización de elementos de sintaxis en la corriente de bits. Esto hace que dichos enfoques basados en RPL tengan un recuento de bits superior para señalización cuando se comparan con otros enfoques de gestión explícita de imágenes de referencia tales como un enfoque basado en RPS. La ineficiencia de la señalización de los enfoques basados en RPL se debe a varios

aspectos.

Por ejemplo, algunos elementos de sintaxis en una estructura RPL son codificados con codificación entrópica menos eficiente. Por ejemplo, la codificación del elemento de sintaxis que representa el valor delta POC de imágenes de referencia a corto plazo (STRP) utiliza un elemento de sintaxis codificado Exp-Golomb de 0-ésimo orden entero con signo con el primer bit izquierdo (por ejemplo,  $se(v)$ ) debido a que el valor delta POC puede tener o bien un valor positivo, o bien un valor negativo. Para codificar cualquier número entero  $x$  no negativo utilizando el código exp-Golomb, la primera etapa es escribir  $x+1$  en binario. A continuación, contar los bits escritos, sustraer uno, y después escribir ese número de inicio de cero bits precediendo a la secuencia de bits previa. Los primeros pocos valores del código son:  $0 \Rightarrow 1 \Rightarrow 1$ ,  $1 \Rightarrow 10 \Rightarrow 010$ ,  $2 \Rightarrow 11 \Rightarrow 011$ ,  $3 \Rightarrow 100 \Rightarrow 00100$ . Sin embargo, puede ser más eficiente si el valor de signo y el valor absoluto del delta POC son codificados separadamente.

La señalización de índices RPL (por ejemplo, para la lista 0 de imágenes de referencia y la lista 1 de imágenes de referencia) cuando se refieren a las estructuras RPL predefinidas en el conjunto de parámetros (por ejemplo, SPS) no puede ser muy eficiente debido a que dichos esquemas siempre señalizan ambos índices. En muchos casos, las estructuras RPL predefinidas para RPL 0 y RPL 1 pueden ser dispuestas de manera que cuando las RPL de una imagen se refieren a las estructuras RPL predefinidas en el conjunto de parámetros, el índice para RPL 0 y el índice para RPL 1 son los mismos.

Además, no hay codificación inter-RPL para escribir una estructura de lista de imágenes de referencia en una corriente de bits de video codificada. Por ejemplo, estructuras RPL pueden ser codificadas sin referencia a estructura o estructuras RPL previamente señalizadas en la corriente de bits. En una realización, una estructura RPL se refiere a una construcción de programación que contiene una lista indexada de indicaciones a imágenes de referencia candidatas.

En la presente memoria se describen varios aspectos, que pueden ser aplicados individualmente y/o en combinación, que resuelven los problemas o abordan los inconvenientes descritos en la presente memoria. Cada uno de esos diversos aspectos se detallan a continuación.

En un aspecto, el valor de signo y el valor absoluto son codificados separadamente en la corriente de bits de video cuando los valores delta POC son codificados para STRP. Para indicar si valores delta POC en cada estructura RPL tienen el mismo valor de signo (por ejemplo, o bien todos los Delta POC en la misma estructura RPL son valores Delta POC positivos o bien son valores Delta POC negativos), un indicador es señalado en el mismo conjunto de parámetros (por ejemplo, SPS, PPS) que las estructuras RPL predefinidas. El indicador puede ser denominado como un "all\_rpl\_entries\_same\_sign\_flag".

Cuando el all\_rpl\_entries\_same\_sign\_flag es igual a 1, un indicador es señalado en la estructura RPL para indicar el valor de signo de todas las entradas (por ejemplo, valor de signo de valores Delta POC de STRP) en la estructura RPL. Cuando el all\_rpl\_entries\_same\_sign\_flag es igual a 0, un indicador es señalado en la estructura RPL para indicar el valor de signo para cada una de las entradas asociada con STRP en cada estructura RPL. En una realización, el valor absoluto de un valor Delta POC de entradas STRP en la estructura RPL es codificado utilizando codificación Exp-Golomb de 0-ésimo orden de entero sin signo (por ejemplo,  $ue(v)$ ).

Alternativamente, en lugar de un indicador para all\_rpl\_entries\_same\_sign\_flag, pueden ser señalizados dos indicadores; uno para cada lista de estructura RPL (por ejemplo, lista de estructura RPL para RPL0 y lista de estructura RPL para RPL1).

Para codificación, cuando se crean estructuras RPL predefinidas para listas RPL 0 y RPL 1 en el conjunto de parámetros, la estructura RPL en la lista para RPL 0 y RPL 1 puede disponerse de manera que la estructura RPL para la lista 0 y la estructura RPL para la lista 1 están emparejadas. Debido a esto, cuando una imagen se refiere a las estructuras RPL predefinidas en el conjunto de parámetros, los índices para RPL0 y RPL0 de esa imagen son los mismos.

Relacionado con el punto anterior, un indicador puede indicar si elementos de sintaxis para RPL 1 que hacen referencia a la estructura RPL predefinida están presentes o no en una cabecera de segmento. El indicador puede ser denominado el "rpl1\_idx\_present\_flag". Este indicador puede ser señalado en SPS o PPS dependiendo del alcance pretendido o la persistencia del indicador. En una realización, se prefiere la señalización del indicador en PPS.

Basado en el valor del rpl1\_idx\_present\_flag, se aplica lo siguiente. Cuando el rpl1\_idx\_present\_flag es igual a 1, si el RPL 1 de la cabecera de segmento asociada con el conjunto de parámetros que contiene el indicador se refiere a la estructura RPL predefinida y su índice no es señalado en la cabecera de segmento y en su lugar es inferido a partir del elemento de sintaxis correspondiente para el RPL 0 en el mismo segmento. Es decir,  $ref\_pic\_list\_sps\_flag[1]$  y  $ref\_pic\_list\_idx[1]$  no son señalizados sino que son inferidos (por ejemplo, copiados) a partir del valor de  $ref\_pic\_list\_sps\_flag[0]$  y  $ref\_pic\_list\_idx[0]$ , respectivamente. De otra manera, cuando el rpl1\_idx\_present\_flag es igual a 0,  $ref\_pic\_list\_sps\_flag[1]$  y  $ref\_pic\_list\_idx[1]$

están presentes en la cabecera de segmento asociada con el conjunto de parámetros que contiene el indicador.

Alternativamente, basándose en el valor de `rpl1_idx_present_flag`, se aplica lo siguiente. Cuando el `rpl1_idx_present_flag` es igual a 1, si la RPL 1 no son señalizados en la cabecera de segmento cuando la RPL 1 de la cabecera de segmento asociada con el conjunto de parámetros que contiene el indicador se refiere a la estructura RPL predefinida. En su lugar, el índice de RPL 1 se deduce del elemento sintáctico correspondiente para la RPL 0 en el mismo segmento. Es decir, `ref_pic_list_sps_flag[ 1 ]` y `ref_pic_list_idx[ 1 ]` no son señalizados sino inferidos (por ejemplo, copiados) a partir del valor de `ref_pic_list_idx[ 0 ]`, respectivamente. De otra manera, cuando el `rpl1_idx_present_flag` es igual a 0, `ref_pic_list_idx[ 1 ]` están presentes en la cabecera de segmento asociada con el conjunto de parámetros que contiene el indicador.

Un indicador puede ser utilizado para indicar si las estructuras RPL en la lista para RPL 1 tienen contenido idéntico que las estructuras RPL en la lista para RPL 0. El indicador puede ser denominado como "`rpl1_copy_from_rpl0_flag`". El indicador es señalizado en el mismo conjunto de parámetros que contiene las estructuras RPL predefinidas y estará ubicado antes de la señalización de las estructuras RPL predefinidas.

Basándose en el valor de `rpl1_copy_from_rpl0_flag`, se aplica lo siguiente. Cuando el `rpl1_copy_from_rpl0_flag` es igual a 1, el número de estructuras RPL para la lista 1 no es señalizado y en su lugar es inferido como siendo el mismo que el número de estructuras RPL para la lista 0. Ninguna estructura RPL para la lista 1 es señalizada. En su lugar, después que las estructuras RPL predefinidas para la lista 0 son analizadas y decodificadas en el decodificador, para cada estructura RPL para la lista 0, se crea una misma copia exacta de la estructura RPL y la copia de la estructura RPL es asignada como estructura RPL para la lista 1 con el mismo índice. De otra manera, cuando el `rpl1_copy_from_rpl0_flag` es igual a 0, el número de estructuras RPL para la lista 1 y de estructuras RPL para la lista 1 es señalizado.

El contenido de una estructura RPL puede ser predicho a partir de otra estructura RPL. Cuando estructuras RPL predefinidas en un conjunto de parámetros se dividen en dos listas (por ejemplo, lista de estructuras RPL para la lista 0 y lista de estructuras RPL para la lista 1), solamente se pueden utilizar estructuras RPL en la lista para lista 0 como referencia para inter-RPL. Una estructura RPL en la lista para lista 0 solamente se puede referir a otra estructura RPL en la misma lista con un índice menor que su índice mientras que una estructura RPL en la lista para lista 1 se puede referir a cualquier estructura RPL en la lista para lista 0. La estructura RPL explícitamente señalizada en la cabecera de segmento se puede referir a cualquiera de las estructuras RPL predefinidas en la lista para lista 0. Alternativamente, una estructura RPL en la lista para lista 0 o lista 1 se puede referir solamente a otra estructura RPL en la lista para lista 0 con un índice menor que su índice. La estructura RPL que es señalizada explícitamente en la cabecera de segmento se puede referir a cualquiera de las estructuras RPL predefinidas en la lista para lista 0.

En una alternativa, cuando estructuras RPL predefinidas en un conjunto de parámetros se dividen en dos listas (por ejemplo, lista de estructuras RPL para la lista 0 y lista de estructuras RPL para la lista 1), una estructura RPL en la lista para lista 0 se puede referir solamente a otra estructura RPL en la lista para lista 0 con un índice menor que su índice. De igual forma, una estructura RPL en la lista para lista 1 se puede referir solamente a otra estructura RPL en la lista para lista 1 con un índice menor que su índice. La estructura RPL explícitamente señalizada en la cabecera de segmento para RPL0 se puede referir a cualquiera de las estructuras RPL predefinidas en la lista para lista 0 y la estructura RPL que es explícitamente señalizada en la cabecera de segmento para RPL1 se puede referir a cualquiera de las estructuras RPL predefinidas en la lista para lista 1.

En otra alternativa, cuando estructuras RPL predefinidas en un conjunto de parámetros no se dividen en dos listas (por ejemplo, lista de estructuras RPL para lista 0 y lista de estructuras RPL para lista 1 son señalizadas en una lista), una estructura RPL se puede referir solamente a otra estructura RPL en la lista con un índice menor que su índice. La estructura RPL señalizada explícitamente en la cabecera de segmento se puede referir a cualquiera de las estructuras RPL predefinidas.

Para inter-RPL, el índice para la estructura RPL de referencia puede ser codificado como el delta del índice de estructura RPL actual y el índice de estructura RPL de referencia menos 1 y codificado con codificación `ue(v)`. En una alternativa, el índice RPL de referencia es codificado directamente utilizando codificación `u(v)`. El número de bits utilizado para representar el índice se establece como  $\log_2$  del número de estructuras RPL en la lista elegible para referencia. Por ejemplo, cuando solamente se pueden usar estructuras RPL en la lista para lista 0 como la referencia, el número de bits para representar el índice RPL de referencia es  $\log_2$  del número de estructuras RPL en la lista para lista 0. En otra alternativa, el índice para la estructura RPL de referencia puede ser codificado utilizando bien `ue(v)` o bien `u(v)` dependiendo del modo de inter-RPL.

Para soportar inter-RPL, cada estructura RPL es codificada utilizando uno de los modos descritos a continuación. Obsérvese que el orden del modo siguiente más adelante no necesariamente significa el orden para el valor del modo. La indicación del modo puede ser codificada de la siguiente manera. En una realización, la indicación del modo puede ser codificada simplemente utilizando `ue(v)`. En una realización, la

indicación del modo puede ser codificada utilizando  $u(v)$  siendo  $\log 2$  el número de bits para representar el modo, del número total de los modos definidos.

5 El primer modo de codificación RPL es un modo de intra-codificación. Este modo funciona igual que los otros mecanismos para señalar el contenido de la estructura RPL. Véase, por ejemplo, el método descrito en la Solicitud Provisional de EE. UU., n.º 62/719,360 presentada el 17 de Agosto, 2018 y titulada "Reference Picture Management in Video Coding" ("Gestión de Imágenes de Referencia en Codificación de Video"). Alternativamente, un valor entero mayor que 0 puede ser señalado para cada estructura RPL y denominado `granularity_val`. El valor de `granularity_val` es utilizado para escalar o dividir cada valor que representa los valores de delta POC de STRP en la estructura RPL.

10 El segundo modo de codificación RPL es un modo de inter-codificación que utiliza los valores delta POC para STRP en la estructura RPL de referencia y la estructura RPL actual que tienen la misma diferencia o una diferencia uniforme. Para codificar RPL utilizando este modo, la siguiente información es señalizada en la corriente de bits.

15 Para comenzar, el modo de codificación y el índice de referencia son señalizados en la corriente de bits. El índice de referencia es el índice de la estructura RPL de referencia. El índice de referencia puede ser codificado como se describió antes. Sin embargo, hay mecanismos que también pueden codificar un índice de referencia como el delta entre el índice de estructura RPL actual y el índice de estructura RPL de referencia menos 1 y codificado como  $ue(v)$ . El desplazamiento también es señalizado en la corriente de bits. El desplazamiento es la diferencia entre el delta POC de la STRP en la estructura RPL de referencia y el  
20 delta POC de la STRP en la estructura RPL actual. El valor del desplazamiento puede estar restringido para que sea solamente positivo (por ejemplo, si el valor delta POC de la STRP en la estructura RPL de referencia es menor que el valor delta POC de la STRP en la estructura RPL actual, entonces este modo no puede ser utilizado para codificar la estructura RPL actual), solamente negativo, o puede ser o bien positivo o bien negativo. El desplazamiento puede ser señalizado con `term_minus1` si está codificado como  $ue(v)$ . Un  
25 indicador para cada entrada en la estructura RPL de referencia para indicar si la entrada es utilizada como una entrada en la estructura RPL actual, también es señalizado en la corriente de bits. Cuando una entrada es una entrada STRP en la estructura RPL de referencia y es utilizada como un entrada en la estructura RPL actual, la entrada también es una entrada STRP en la estructura RPL actual y su valor es el valor de entrada en la estructura RPL de referencia menos el desplazamiento (más 1 `if_minus1` term es utilizado para codificar el desplazamiento). Cuando una entrada es una entrada de imagen de referencia a largo plazo (LTRP) en la  
30 estructura RPL de referencia y es utilizada como una entrada en la estructura RPL actual, la entrada es también una entrada LTRP en la estructura RPL actual y su valor simplemente es copiado para la entrada en la estructura RPL de referencia.

35 El tercer modo de codificación RPL es un modo de inter-codificación que utiliza el hecho de que entradas (por ejemplo, STRP y LTRP) en la estructura RPL actual son un súper conjunto de entradas en la estructura RPL de referencia. Es decir, todas las entradas en la estructura RPL de referencia son las mismas que la primera X (reemplazar X con el número de entradas en la estructura RPL de referencia) con cero o más entradas adicionales. Para codificar RPL utilizando este modo, la siguiente información es señalizada en la corriente de bits.

40 Para comenzar, el modo de codificación y el índice de referencia son señalizados en la corriente de bits. El índice de referencia es el índice de la estructura RPL de referencia. El índice de referencia puede ser codificado como se describió antes. El número de entradas adicionales también es señalizado en la corriente de bits. El número de entradas adicionales es la diferencia entre el número de entradas en la estructura RPL actual y el número de entradas en la estructura RPL de referencia. Cuando una entrada es una entrada  
45 STRP en la estructura RPL de referencia, la entrada también es una entrada STRP en la estructura RPL actual y su valor simplemente es copiado para la entrada en la estructura RPL de referencia. Cuando una entrada es una entrada LTRP en la estructura RPL de referencia, la entrada también es una entrada LTRP en la estructura RPL actual y su valor es simplemente copiado para la entrada en la estructura RPL de referencia. Después de que todas las entradas en la estructura RPL de referencia son copiadas en la  
50 estructura RPL actual, para cada entrada adicional, se señala la siguiente información: Si imagen de referencia a largo plazo es utilizada en la corriente de bits (es decir, esto puede ser indicado mediante un indicador en el mismo conjunto de parámetros), un indicador es señalizado para indicar si la entrada adicional es una entrada LTRP o una entrada STRP. Si la entrada es una entrada LTRP, se señala el POC LSB de la entrada LTRP. De otra manera, se señala el delta POC de la entrada STRP. El valor de delta POC puede  
55 ser señalizado como el delta desde la entrada STRP anterior o simplemente el delta desde el POC de la imagen actual.

60 El cuarto modo de codificación RPL es un modo de inter-codificación que utiliza el hecho de que las entradas (por ejemplo, STRP y LTRP) en la estructura RPL actual son, o bien exactamente las mismas que las entradas en la estructura RPL de referencia, o bien exactamente las mismas con el valor de signo dado la vuelta. Para codificar las RPL utilizando este modo, se señala la siguiente información en la corriente de bits.

Para comenzar, el modo de codificación y el índice de referencia son señalizados. El índice de referencia es el índice de la estructura RPL de referencia. El índice de referencia puede ser codificado como se describió antes. Opcionalmente, un indicador para indicar si el valor de signo está dado la vuelta también es señalizado en la corriente de bits.

- 5 Cuando una corriente de bits es codificada con inter-predicción de avance (por ejemplo, inter-predicción con referencia a una imagen con valor POC menor que el valor POC de la imagen actual) e inter-predicción de retroceso (por ejemplo, inter-predicción con referencia a una imagen con valor POC mayor que el valor POC de la imagen actual), se aplican las siguientes restricciones para imágenes de referencia en una RPL. Para cada RPL de una imagen, todas las imágenes de referencia en la RPL son de la misma dirección de inter-predicción, por ejemplo, todas las imágenes de referencia son imágenes de referencia para inter-predicción de avance o todas las imágenes de referencia son imágenes de referencia para inter-predicción de retroceso. Para el par de RPL de una imagen, no habrá una imagen de referencia contenida en ambas RPL0 y RPL1 a menos que sea verdadera la siguiente condición. La condición es que tanto RPL0 como RPL1 contienen las mismas imágenes de referencia con el mismo orden. En dicha condición, todas las imágenes de referencia, excepto la primera (es decir, aquella en el índice más bajo) en la RPL1, son eliminadas de RPL1. Alternativamente, todas las imágenes de referencia, excepto la primera (es decir, aquella en el índice más bajo) en la RPL0, son eliminadas de RPL0.

20 En la presente memoria se han descrito técnicas de codificación de video que utilizan un indicador para especificar si candidatos para una segunda estructura de lista de imágenes de referencia pueden ser inferidos para que sean los mismos que candidatos para una primera estructura de lista de imágenes de referencia. Es decir, los candidatos para la segunda estructura de lista de imágenes de referencia no están presentes en la corriente de bits de video codificada y son inferidos para que sean los mismos que los candidatos para la primera estructura de lista de imágenes de referencia cuando el indicador tiene un primer valor. Por otra parte, los candidatos para la segunda estructura de lista de imágenes de referencia están presentes en la corriente de bits de video codificada cuando el indicador tiene un segundo valor. Al utilizar el indicador de esta manera, el codificador/decodificador (también conocido como "códec") en la codificación de video se mejora (por ejemplo, utiliza menos bits, demanda menos ancho de banda, es más eficiente, etc.) con relación a los códecs actuales. Como un asunto práctico, el proceso de codificación de video mejorado ofrece al usuario una mejor experiencia de usuario cuando se envían, reciben y/o visualizan videos.

30 En una realización, un candidato es uno de una lista de vectores de movimiento disponibles para realizar la inter-predicción para un bloque correspondiente. Durante la codificación, un codificador genera una lista de candidatos de vector de movimiento para inter-predicción, selecciona el candidato que da como resultado la mejor compresión, y señala el candidato seleccionado por el índice de estructura RPL.

35 La figura 5 es un diagrama esquemático de una realización de una corriente 500 de bits de video. Tal como se utiliza en la presente memoria, la corriente 500 de bits de video también se puede denominar como una corriente de bits de video codificada, una corriente de bits, o variaciones de la misma. Como se muestra en la figura 5, la corriente 500 de bits comprende un conjunto de parámetros de secuencia (SPS) 510, un conjunto de parámetros de imagen (PPS) 512, una cabecera 514 de segmento, y datos 520 de imagen.

40 El SPS 510 contiene datos que son comunes para todas las imágenes en una secuencia de imágenes (SOP). En contraste, el PPS 512 contiene datos que son comunes a la imagen completa. La cabecera 514 de segmento contiene información acerca del segmento actual tal como, por ejemplo, el tipo de segmento, cuáles de las imágenes de referencia serán utilizadas, y así sucesivamente. El SPS 510 y el PPS 512 genéricamente se pueden denominar como un conjunto de parámetros. El SPS 510, el PPS 512, y la cabecera 514 de segmento son tipos de unidades de Capa de Abstracción de Red (NAL). Los datos 520 de imagen comprenden datos asociados con las imágenes o video que está siendo codificado o decodificado. Los datos 520 de imagen simplemente pueden ser denominados como la carga útil o datos que son llevados en la corriente 500 de bits.

50 En una realización, el SPS 510, el PPS 512, la cabecera 514 de segmento, u otra porción de la corriente 500 de bits llevan una pluralidad de estructuras de lista de imágenes de referencia, cada una de las cuales contiene una pluralidad de entradas de imagen de referencia. Los expertos en la técnica apreciarán que la corriente 500 de bits puede contener otros parámetros e información en aplicaciones prácticas.

60 La figura 6 es una realización de un método 600 de decodificación de una corriente de bits de video codificada (por ejemplo, corriente 500 de bits) implementada por un decodificador de video (por ejemplo, decodificador 30 de video). El método 600 puede ser realizado después que la corriente de bits decodificada ha sido recibida directamente o indirectamente desde un codificador de video (por ejemplo, codificador 20 de video). El método 600 mejora el proceso de decodificación (por ejemplo, hace que el proceso de decodificación sea más eficiente, más rápido, etc.), que los procesos de decodificación convencionales) debido a que candidatos para una segunda estructura de lista de imágenes de referencia pueden ser inferidos como iguales que candidatos para la primera estructura de lista de imágenes de referencia cuando un indicador se establece a un cierto valor. Es decir, la segunda estructura de lista de imágenes de referencia

no necesita ser señalizada en la corriente de bits codificada en cada circunstancia como en HEVC y AVC. Por lo tanto, como un asunto práctico, se mejora el rendimiento de un códec, lo cual conduce a una mejor experiencia de usuario.

5 En el bloque 602, un indicador es analizado procedente de la corriente de bits de video codificada (por ejemplo, la corriente 500 de bits de video). En una realización, el indicador es designado `rpl1_copy_from_rpl0_flag`. En una realización, el indicador es incluido en el SPS (por ejemplo, SPS 510) de la corriente de bits de video codificada. En una realización, el primer valor del indicador es uno (1). En una realización, el segundo valor del indicador es cero (0). En una realización, el indicador es dispuesto en la corriente de bits de video codificada antes que cualquier estructura de lista de imágenes de referencia  
10 predefinida.

15 En el bloque 604, los candidatos para una primera estructura de lista de imágenes de referencia son analizados a partir de la corriente de bits de video codificada. En una realización, candidatos para la primera estructura de lista de imágenes de referencia son incluidos en el SPS (por ejemplo, el SPS 510) de la corriente de bits de video codificada. En una realización, el indicador y los candidatos para la primera estructura de lista de imágenes de referencia son analizados separadamente uno de otro. Es decir, primero se analiza el indicador, y después se analizan los candidatos para la primera estructura de lista de imágenes de referencia, o viceversa.

20 En el bloque 606, se determina que los candidatos para una segunda estructura de lista de imágenes de referencia no están presentes en la corriente de bits de video codificada y los candidatos para la segunda estructura de lista de imágenes de referencia se infieren como los mismos que los candidatos para la primera estructura de lista de imágenes de referencia cuando el indicador tiene un primer valor. En una realización, los candidatos para la segunda estructura de lista de imágenes de referencia no están presentes en el SPS (por ejemplo, el SPS 510) de la corriente de bits de video codificada cuando el indicador tiene el primer valor. En el bloque 608, se determina que los candidatos para la segunda estructura de lista de imágenes de referencia estén presentes en la corriente de bits de video codificada cuando el indicador tiene un segundo valor.  
25

30 En el bloque 610, se genera una lista de imágenes de referencia utilizando los candidatos para la primera estructura de lista de imágenes de referencia, los candidatos para la segunda estructura de lista de imágenes de referencia, o alguna combinación de las mismas. La lista de imágenes de referencia puede identificar una o más imágenes tal como, por ejemplo, las imágenes ilustradas y descritas en conexión con la figura 4.

En el bloque 612, se realiza inter-predicción basándose en la lista de imágenes de referencia para generar un bloque reconstruido. En una realización, se puede utilizar el bloque reconstruido para generar o producir una imagen para mostrar a un usuario en el dispositivo de visualización o pantalla de un dispositivo electrónico (por ejemplo, un teléfono inteligente, tableta, portátil, ordenador personal, etc.).

35 La figura 7 es una realización de un método 700 de codificación de una corriente de bits de video (por ejemplo, corriente 500 de bits) implementada por un codificador de video (por ejemplo, codificador 20 de video). El método 700 puede ser realizado cuando una imagen (por ejemplo, procedente de un video) ha de ser codificada en una corriente de bits de video y luego transmitida hacia un decodificador de video (por ejemplo, decodificador 30 de video). El método 700 mejora el proceso de codificación (por ejemplo, hace que el proceso de codificación sea más eficiente, más rápido, etc., que los procesos de codificación convencionales) debido a que candidatos para una segunda estructura de lista de imágenes de referencia pueden ser inferidos como los mismos que los candidatos para la primera estructura de lista de imágenes de referencia cuando un indicador es establecido a un cierto valor. Es decir, la segunda estructura de lista de imágenes de referencia no necesita ser señalizada en la corriente de bits codificada en cada circunstancia como en HEVC y AVC. Por lo tanto, como un asunto práctico, se mejora el rendimiento de un códec, lo que conduce a una mejor experiencia de usuario.  
40  
45

50 En el bloque 702, un indicador con un primer valor es codificado en la corriente de bits de video cuando candidatos para una segunda estructura de lista de imágenes de referencia no son codificados en la corriente de bits de video y los candidatos para la segunda estructura de lista de imágenes de referencia deberían ser inferidos por un decodificador de video para que sean los mismos que los candidatos para una primera estructura de lista de imágenes de referencia. En una realización, el indicador es designado como `rpl1_copy_from_rpl0_flag`. En una realización, el primer valor del indicador es uno (1). En una realización, el indicador es codificado en el SPS (por ejemplo, SPS 510) de la corriente de bits de video codificada (por ejemplo, corriente 500 de bits de video). En una realización, los candidatos para la segunda estructura de lista de imágenes de referencia no están presentes en el SPS de la corriente de bits de video codificada cuando el indicador tiene el primer valor. En una realización, el indicador es codificado en la corriente de bits de video codificada antes que cualquier estructura de lista de imágenes de referencia predefinida.  
55

En el bloque 704, un indicador con un segundo valor es codificado en la corriente de bits de video cuando los candidatos para la segunda estructura de lista de imágenes de referencia son codificados en la corriente de

bits de video. En una realización, el segundo valor del indicador es cero (0). En una realización, los candidatos para la segunda estructura de lista de imágenes de referencia están presentes en el SPS de la corriente de bits codificada de video cuando el indicador tiene el primer valor.

5 En el bloque 706, la primera estructura de lista de imágenes de referencia es codificada en la corriente de bits de video cuando el indicador es codificado con el primer valor. En el bloque 708, la primera estructura de lista de imágenes de referencia y la segunda estructura de lista de imágenes de referencia son codificadas en la corriente de bits de video cuando el indicador es codificado con el segundo valor.

10 En el bloque 710, la corriente de bits de video es transmitida hacia el decodificador de video (por ejemplo, el decodificador 30 de video). Una vez recibida por el decodificador de video, la corriente de bits de video codificada puede ser decodificada (por ejemplo, como se describió antes) para generar o producir una imagen para mostrar a un usuario en el dispositivo de visualización o pantalla de un dispositivo electrónico (por ejemplo, un teléfono inteligente, tableta, portátil, ordenador personal, etc.).

15 Una descripción de las técnicas descritas en la presente memoria es proporcionada con relación a la última VVC VVD. Además, también se proporcionan definiciones, sintaxis, y semánticas adecuadas para implementar las técnicas descritas en la presente memoria.

20 Para comenzar, se proporcionan varias definiciones. Una imagen de punto de acceso intra-aleatorio (IRAP) es una imagen codificada para la cual cada unidad NAL de capa de codificación de video (VCL) tiene nal\_unit\_type igual a IRAP\_NUT. Una imagen de referencia a largo plazo (LTRP) es una imagen marcada como "utilizada para referencia a largo plazo." Una imagen no-IRAP es una imagen codificada para la cual cada unidad VCL NAL tiene nal\_unit\_type igual a NON\_IRAP\_NUT. Una lista de imágenes de referencia es una lista de imágenes de referencia utilizadas para inter-predicción de un segmento P o un segmento B. Dos listas de imágenes de referencia, lista 0 de imágenes de referencia y lista 1 de imágenes de referencia, son generadas para cada segmento de una imagen no-IRAP. El conjunto de imágenes únicas referido por todas las entradas en las dos listas de imágenes de referencia asociadas con una imagen incluye todas las  
25 imágenes de referencia que pueden ser utilizadas para inter-predicción de la imagen asociada o cualquier imagen después de la imagen asociada en el orden de decodificación. Para decodificar los datos de segmento de un segmento P, solamente se utiliza la lista 0 de imágenes de referencia para inter-predicción. Para decodificar los datos de segmento de un segmento B, ambas listas de imágenes de referencia son utilizadas para inter-predicción. Para decodificar los datos de segmento de un segmento I, no se utiliza ninguna lista de imágenes de referencia para inter-predicción. Una imagen de referencia a corto plazo (STRP) es una imagen marcada como "utilizada para referencia a corto plazo."  
30

A continuación se proporcionan algunas abreviaturas. Tal como aquí se utiliza, LTRP representa Imagen de Referencia a Largo Plazo y STRP representa Imagen de Referencia a Corto Plazo.

35 Las siguientes porciones de la descripción proporcionan la sintaxis y semántica adecuadas para implementar las técnicas descritas en la presente memoria.

Sintaxis de cabecera 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)
}	

Sintaxis de conjunto de parámetros de secuencia RBSP

40

seq_parameter_set_rbsp( ) {	Descriptor
sps_seq_parameter_set_id	ue(v)

## ES 2 981 234 T3

chroma_format_idc	ue(v)
si( chroma_formato_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)
si( long_term_ref_pics_flag )	
additional_lt_poc_lsb	ue(v)
all_rpl_entries_same_sign_flag	u(1)
rpl1_copy_from_rpl0_flag	u(1)
para( i = 0; i < ( !rpl1_copy_from_rpl0_flag ) ? 2 : 1; i++ ) {	
num_ref_pic_lists_in_sps[ i ]	ue(v)
para( 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( )	
}	

Sintaxis de conjunto de parámetros de imagen Rbsp

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

## ES 2 981 234 T3

rpl1_idx_present_flag	u(1)
rbsp_trailing_bits( )	
}	

Sintaxis de cabecera de segmento

slice_header( ) {	Descriptor
slice_pic_parameter_set_id	ue(v)
slice_address	u(v)
slice_type	ue(v)
si( slice_type != I )	
log2_diff_ctu_max_bt_size	ue(v)
si( nal_unit_type != IRAP_NUT ) {	
slice_pic_order_cnt_lsb	u(v)
para( i = 0; i < 2; i++ ) {	
si( i == 0    ( i == 1 && rpl1_idx_present_flag ) )	
ref_pic_list_sps_flag[ i ]	u(1)
si( ref_pic_list_sps_flag[ i ] ) {	
si( num_ref_pic_lists_in_sps[ i ] > 1 )	
si( i == 0    ( i == 1 && rpl1_idx_present_flag ) )	
ref_pic_list_idx[ i ]	u(v)
} además	
ref_pic_list_struct( i, num_ref_pic_lists_in_sps[ i ], long_term_ref_pics flag	
}	
si( slice_type == P    slice_type == B ) {	
num_ref_idx_active_override_flag	u(1)
si( num_ref_idx_active_override_flag )	
para( i = 0; i < ( slice_type == B ? 2 : 1 ); i++ )	
num_ref_idx_active_minus1[ i ]	ue(v)
}	
}	
byte_alignement( )	
}	

## ES 2 981 234 T3

Sintaxis de estructura de lista de imágenes de referencia

	Descriptor
ref_pic_list_struct( listIdx, rplIdx, ltrpFlag ) {	
rpl_mode[ listIdx ][ rplIdx ]	u(2)
si( rpl_mode[ listIdx ][ rplIdx ] == 0 ) {	
num_strp_entries[ listIdx ][ rplIdx ]	ue(v)
si( ltrpFlag )	
num_ltrp_entries[ listIdx ][ rplIdx ]	ue(v)
si( all_rpl_entries_same_sign_flag )	
strp_entries_sign_flag[listIdx][rplIdx]	u(1)
para( i = 0; i < NumEntriesInList[ listIdx ][ rplIdx ]; i++) {	
si( num_ltrp_entries[ listIdx ][ rplIdx ] > 0 )	
lt_ref_pic_flag[ listIdx ][ rplIdx ][ i ]	u(1)
si( !lt_ref_pic_flag[ listIdx ][ rplIdx ][ i ] ) {	
si( !all_rpl_entries_same_sign_flag )	
strp_entry_sign_flag[listIdx][rplIdx][ i ]	u(1)
delta_poc_st[ listIdx ][ rplIdx ][ i ]	ue(v)
} además	
poc_lsb_lt[ listIdx ][ rplIdx ][ i ]	u(v)
}	
} además si( rpl_mode[ listIdx ][ rplIdx ] == 1 ) {	
mode1_ref_rpl_idx_delta_minus1[ listIdx ][ rplIdx ]	ue(v)
strp_offset_val_minus1[ listIdx ][ rplIdx ]	ue(v)
para( i = 0; i < NumEntriesInList[ 0 ][ RefRplIdx ]; i++)	
ref_entrada_utilizado_indicador[ listIdx ][ rplIdx ][ i ]	u(1)
} además si( rpl_mode[ listIdx ][ rplIdx ] == 2 ) {	
mode2_ref_rpl_idx[ listIdx ][ rplIdx ]	u(v)
num_additional_entries[ listIdx ][ rplIdx ]	ue(v)
para( i = 0; i < num_additional_entries[ listIdx ][ rplIdx ]; i++) {	
si( ltrpFlag )	
add_lt_ref_pic_flag[ listIdx ][ rplIdx ][ i ]	u(1)
si( !add_lt_ref_pic_flag[ listIdx ][ rplIdx ][ i ] ) {	
si( !all_rpl_entries_same_sign_flag )	
add_strp_entry_sign_flag[listIdx][rplIdx][ i ]	u(1)
add_delta_poc_st[ listIdx ][ rplIdx ][ i ]	ue(v)

} además	
add_poc_lsb_l[ listIdx ][ rplIdx ][ i ]	u(v)
}	
} además si( rpl_mode[ listIdx ][ rplIdx ] = 3 )	
modo3_ref_rpl_idx[ listIdx ][ rplIdx ]	u(v)
}	

Semántica de cabecera 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.

5 Tabla 4-1 - 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 sintáctica de RBSP	Clase tipo unidad NAL
0	NO_IRAP_NUT	Segmento codificado de una imagen no-IRAP slice_layer_rbsp( )	VCL
1	IRAP_NUT	Segmento codificado de una imagen IRAP slice_layer_rbsp( )	VCL
2-15	RSV_VCL_NUT	Unidades VCL NAL 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 corriente 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	Reservado	no-VCL
27-31	UNSPEC	No especificado	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 de la siguiente manera:

10 TemporalId = nuh\_temporal\_id\_plus1 - 1. Cuando nal\_unit\_type es igual a IRAP\_NUT, el segmento codificado pertenece a una imagen IRAP, TemporalId será igual a 0. El valor de TemporalId será el mismo para todas las unidades VCL NAL de una unidad de acceso. El valor de TemporalId de una imagen codificada o una unidad de acceso es el valor del TemporalId de las unidades VCL NAL de la imagen codificada o la unidad de acceso. El valor de TemporalId para unidades no-VCL NAL es restringido de la

15 siguiente manera: Si nal\_unit\_type es igual a SPS\_NUT, TemporalId será igual a 0 y el TemporalId de la unidad de acceso que contiene la unidad NAL será igual a 0. De otra manera si nal\_unit\_type es igual a EOS\_NUT o EOB\_NUT, TemporalId será igual a 0. De otra manera, TemporalId será mayor que o igual al TemporalId de la unidad de acceso que contiene la unidad NAL. Cuando la unidad NAL es una unidad no-VCL NAL, el valor de TemporalId es igual al valor mínimo de los valores TemporalId de todas las unidades de

20 acceso a las que se aplica la unidad no-VCL NAL. Cuando nal\_unidad\_tipo es igual a PPS\_NUT, TemporalId puede ser mayor que o igual al TemporalId de la unidad de acceso, dado que todos los conjuntos de

parámetros de imagen (PPS) pueden ser incluidos al inicio de una corriente 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 puede ser mayor que o igual al TemporalId de la unidad de acceso que contiene, dado que una unidad SEI NAL puede contener información que se aplica a un subconjunto de corriente de bits que incluye las unidades de acceso para las cuales los valores TemporalId son mayores que el TemporalId de la unidad de acceso que contiene la unidad SEI NAL. nuh\_reserved\_zero\_7bits será igual a '0000000'. Otros valores de nuh\_reserved\_zero\_7bits pueden ser especificados en el futuro por ITU-T | ISO/IEC. Los decodificadores ignorarán (es decir, eliminar de la corriente de bits y desechar) unidades NAL con valores de nuh\_reserved\_zero\_7bits no iguales a '0000000'.

10 Semántica de conjunto de parámetros de secuencia RBSP

Un log2\_max\_pic\_order\_cnt\_lsb\_minus4 especifica el valor de la variable MaxPicOrderCntLsb que se utiliza en el proceso de decodificación para recuento de orden de imágenes del siguiente modo:  $MaxPicOrderCntLsb = 2(\log2\_max\_pic\_order\_cnt\_lsb\_minus4 + 4)$ . El valor log2\_max\_pic\_order\_cnt\_lsb\_minus4 será del orden de 0 a 12, inclusive. El valor log2\_max\_pic\_order\_cnt\_lsb\_minus4 será del orden de 0 a 12, inclusive. sps\_max\_dec\_pic\_buffering\_minus1 plus 1 especifica el tamaño máximo requerido de la memoria tampón de imágenes decodificadas para la secuencia de video codificada (CVS) en unidades de memorias tampón de almacenamiento de imágenes. El valor de sps\_max\_dec\_pic\_buffering\_minus1 será del orden de 0 a MaxDpbSize - 1, inclusive, donde MaxDpbSize es como se especificó además en alguna otra parte. long\_term\_ref\_pics\_flag igual a 0 especifica que no se ha utilizado ninguna LTRP para inter-predicción de ninguna imagen codificada en el CVS. long\_term\_ref\_pics\_flag igual a 1 especifica que se pueden utilizar LTRP para inter-predicción de una o más imágenes codificadas en el CVS. additional\_lt\_poc\_lsb especifica el valor de la variable MaxLtPicOrderCntLsb que se utiliza en el proceso de decodificación para 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 será del orden 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 infiere como igual a 0.

Un all\_rpl\_entries\_same\_sign\_flag igual a 1 especifica que todas las entradas de STRP en cada ref\_pic\_list\_struct( listIdx, rplIdx, ltrpFlag ) tienen el mismo valor de signo, donde el signo indica valores positivos o negativos. all\_rpl\_entries\_same\_sign\_flag igual a 0 especifica que entradas de STRP en cada ref\_pic\_list\_struct( listIdx, rplIdx, ltrpFlag ) pueden o no tener el mismo valor de signo. rpl1\_copy\_from\_rpl0\_flag igual a 1 especifica que num\_ref\_pic\_lists\_in\_sps[ 1 ] y ref\_pic\_list\_struct( 1, rplIdx, ltrpFlag ) no están presentes y se aplica lo siguiente: El valor de num\_ref\_pic\_lists\_in\_sps[ 1 ] se establece igual al valor de num\_ref\_pic\_lists\_in\_sps[ 0 ]. La estructura sintáctica ref\_pic\_list\_struct( 1, rplIdx, ltrpFlag ) es inferida como idéntica al valor de ref\_pic\_list\_struct( 0, rplIdx, ltrpFlag ). Por consiguiente, los elementos sintácticos en ref\_pic\_list\_struct( 1, rplIdx, ltrpFlag ) se infieren respectivamente como iguales a los elementos de sintaxis en ref\_pic\_list\_struct( 0, rplIdx, ltrpFlag ).

Un num\_ref\_pic\_lists\_in\_sps[ i ] especifica el número de las estructuras sintácticas ref\_pic\_list\_struct( listIdx, rplIdx, ltrpFlag ) con listIdx igual a i incluido en el SPS. El valor de num\_ref\_pic\_lists\_in\_sps[ i ] será del orden de 0 a 64, inclusive. Para cada valor de listIdx (igual a 0 o 1), un decodificador debería asignar memoria para un número total de sintácticas num\_ref\_pic\_lists\_in\_sps[ i ] + 1 ref\_pic\_list\_struct( listIdx, rplIdx, ltrpFlag ) estructuras sintácticas dado que puede haber una estructura sintáctica ref\_pic\_list\_struct( listIdx, rplIdx, ltrpFlag ) directamente señalizada en las cabeceras de segmento de una imagen actual.

Semántica de conjunto de parámetros de imagen RBSP

Un num\_ref\_idx\_default\_active\_minus1[ i ] plus 1, cuando i es igual a 0, especifica el valor inferido de la variable NumRefIdxActive[ 0 ] para segmentos P o B con num\_ref\_idx\_active\_override\_flag igual a 0, y, cuando i es igual a 1, especifica el valor inferido de NumRefIdxActive[ 1 ] para segmentos B con num\_ref\_idx\_active\_override\_flag igual a 0. El valor de num\_ref\_idx\_default\_active\_minus1 [ i ] será del orden de 0 a 14, inclusive. rpl1\_idx\_present\_flag igual a 0 especifica que ref\_pic\_list\_sps\_flag[ 1 ] y ref\_pic\_list\_idx[ 1 ] no están presentes en las cabeceras de segmento. rpl1\_idx\_present\_flag igual a 1 especifica que ref\_pic\_list\_sps\_flag[ 1 ] y ref\_pic\_list\_idx[ 1 ] puede estar presente en cabeceras de segmento.

Semántica de cabecera de segmento

Cuando está presente, el valor de cada uno de los elementos de sintaxis de cabecera de segmento slice\_pic\_parameter\_set\_id y slice\_pic\_order\_cnt\_lsb serán los mismos en todas las cabeceras de segmento de una imagen codificada. ... slice\_type especifica el tipo de codificación del segmento de acuerdo con la tabla 7-3.

Tabla 7-3 – Asociación de nombre con slice\_type

slice_type	Nombre de slice_type
0	B (segmento B)
1	P (segmento P)
2	I (segmento 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\_type\_pic\_order\_cnt\_lsb especifica el módulo MaxPicOrderCntLsb de recuento de orden de imágenes 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 del slice\_pic\_order\_cnt\_lsb será del orden de 0 a MaxPicOrderCntLsb - 1, inclusive. Cuando slice\_pic\_order\_cnt\_lsb no está presente, slice\_pic\_order\_cnt\_lsb se infiere como 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 es derivada basándose en una de las estructuras sintácticas de ref\_pic\_list\_struct ( listIdx, rplIdx, ltrpFlag ) con listIdx igual a i 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 es derivada basándose en la estructura sintáctica de ref\_pic\_list\_struct( listIdx, rplIdx, ltrpFlag ) con listIdx igual a i que está directamente incluida en las cabeceras de segmento 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. Cuando rpl1\_idx\_present\_flag es igual a 0 y ref\_pic\_list\_sps\_flag[ 0 ] está presente, el valor de ref\_pic\_list\_sps\_flag[ 1 ] es inferido como igual al valor de ref\_pic\_list\_sps\_flag[ 0 ]. ref\_pic\_list\_idx[ i ] especifica el índice, dentro de la lista de las estructuras sintácticas de ref\_pic\_list\_struct( listIdx, rplIdx, ltrpFlag ) con listIdx igual a i incluida en el SPS activo, de la estructura sintáctica de ref\_pic\_list\_struct( listIdx, rplIdx, ltrpFlag ) con listIdx igual a i que se utiliza para derivación de lista i de imágenes de referencia de la imagen actual. El elemento de sintaxis ref\_pic\_list\_idx[ i ] es representado por Ceil( Log2( num\_ref\_pic\_lists\_en\_sps[ i ] ) ) bits. Cuando no está presente, el valor de ref\_pic\_list\_idx[ i ] es inferido como igual a 0. El valor de ref\_pic\_list\_idx[ i ] será del orden de 0 a num\_ref\_pic\_lists\_en\_sps[ i ] - 1, inclusive. Cuando rpl1\_idx\_present\_flag es igual a 0 y ref\_pic\_list\_sps\_flag[ 0 ] está presente, el valor de ref\_pic\_list\_idx[ 1 ] se infiere como igual al valor de ref\_pic\_list\_idx[ 0 ]. num\_ref\_idx\_active\_override\_flag igual a 1 especifica que el elemento de sintaxis de num\_ref\_idx\_active\_minus1[ 0 ] está presente para los segmentos P y B y que el elemento de sintaxis de num\_ref\_idx\_active\_minus1[ 1 ] está presente para los segmentos 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 ] será del orden de 0 a 14, inclusive.

35 El valor de NumRefIdxActive[ i ] - 1 especifica el máximo índice de referencia para lista i de imágenes de referencia que pueden ser utilizadas para decodificar el segmento. Cuando el valor de NumRefIdxActive[ i ] es igual a 0, no se puede utilizar ningún índice de referencia para lista i de imágenes de referencia para decodificar el segmento. Para i igual a 0 o 1, cuando el segmento actual es un segmento B y num\_ref\_idx\_active\_override\_flag es igual a 0, NumRefIdxActive[ i ] se infiere como igual a num\_ref\_idx\_default\_active\_minus1[ i ] + 1. Cuando el segmento actual es un segmento P y num\_ref\_idx\_active\_override\_flag es igual a 0, NumRefIdxActive[ 0 ] se infiere como igual a num\_ref\_idx\_default\_active\_minus1[ 0 ] + 1. Cuando el segmento actual es un segmento P, NumRefIdxActive[ 1 ] se infiere como igual a 0. Cuando el segmento actual es un segmento I, ambos NumRefIdxActive[ 0 ] y NumRefIdxActive[ 1 ] se infieren como iguales a 0.

45 Alternativamente, para i igual a 0 o 1, se aplica lo siguiente después de lo anterior: Asumir que rplIdx1 sea establecido igual a ref\_pic\_list\_sps\_flag[ i ]? ref\_pic\_list\_idx[ i ]: num\_ref\_pic\_lists\_in\_sps[ i ], y numRpEntries[ i ] es 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 ] es establecido igual a numRpEntries[ i ].

Semántica de estructura de lista de imágenes de referencia

50 La estructura sintáctica ref\_pic\_list\_struct ( listIdx, rplIdx, ltrpFlag ) puede estar presente en un SPS o en una cabecera de segmento. Dependiendo de si la estructura sintáctica está incluida en una cabecera de segmento o en un SPS, se aplica lo siguiente: Si está presente en una cabecera de segmento, la estructura sintáctica ref\_pic\_list\_struct ( listIdx, rplIdx, ltrpFlag ) especifica la lista de imágenes de referencia listIdx de la imagen actual (la imagen que contiene el segmento). De otra manera (presente en un SPS), la estructura

sintáctica `ref_pic_list_struct ( listIdx, rplIdx, ltrpFlag )` especifica un candidato para la lista de imágenes de referencia `listIdx`, y el término "la imagen actual" en la semántica especificada en el resto de esta sección se refiere a cada imagen que 1) tiene uno o más segmentos que contienen `ref_pic_list_idx[ listIdx ]` igual a un índice en la lista de las estructuras sintácticas `ref_pic_list_struct ( listIdx, rplIdx, ltrpFlag )` incluidas en el SPS, y 2) está en un CVS que tiene el SPS como el SPS activo. `rpl_mode[ listIdx ][ rplIdx ]` especifica el modo de codificación de elementos de sintaxis en la estructura sintáctica `ref_pic_list_struct( listIdx, rplIdx, ltrpFlag )`. `num_strp_entries[ listIdx ][ rplIdx ]` especifica el número de entradas de STRP en la estructura sintáctica `ref_pic_list_struct( listIdx, rplIdx, ltrpFlag )`. `num_ltrp_entries[ listIdx ][ rplIdx ]` especifica el número de entradas de LTRP en la estructura sintáctica `ref_pic_list_struct( listIdx, rplIdx, ltrpFlag )`. Cuando no está presente, el valor de `num_ltrp_entries[ listIdx ][ rplIdx ]` se infiere como igual a 0. La variable `NumEntriesInList[ listIdx ][ rplIdx ]` se deriva de la siguiente manera: `NumEntriesInList[ listIdx ][ rplIdx ] = num_strp_entries[ listIdx ][ rplIdx ] + num_ltrp_entries[ listIdx ][ rplIdx ]`. El valor de `NumEntriesInList[ listIdx ][ rplIdx ]` será del orden de 0 a `sps_max_dec_pic_buffering_minus1`, inclusive. `strp_entries_sign_flag[ listIdx ][ rplIdx ]` igual a 1 especifica que todas las entradas de STRP en `ref_pic_list_struct( listIdx, rplIdx, ltrpFlag )` tienen un valor mayor que 0 igual a 0. `strp_entries_sign_flag[ listIdx ][ rplIdx ]` igual a 0 especifica que todas las entradas de STRP en `ref_pic_list_struct( listIdx, rplIdx, ltrpFlag )` tienen un valor menor que 0.

El `lt_ref_pic_flag[ listIdx ][ rplIdx ][ i ]` igual a 1 especifica que la *i*-ésima entrada en la estructura sintáctica `ref_pic_list_struct ( listIdx, rplIdx, ltrpFlag )` es una entrada de LTRP. `lt_ref_pic_flag[ listIdx ][ rplIdx ][ i ]` igual a 0 especifica que la *i*-ésima entrada en la estructura sintáctica `ref_pic_list_struct( listIdx, rplIdx, ltrpFlag )` es una entrada de STRP. Cuando no está presente, el valor de `lt_ref_pic_flag[ listIdx ][ rplIdx ][ i ]` se infiere como igual a 0. Es un requerimiento de conformidad de corriente 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, será igual a `num_ltrp_entries[ listIdx ][ rplIdx ]`. `strp_entry_sign_flag[ listIdx ][ rplIdx ][ i ]` igual a 1 especifica que la *i*-ésima entrada en `ref_pic_list_struct( listIdx, rplIdx, ltrpFlag )` tiene un valor mayor que 0 igual a 0. `strp_entries_sign_flag[ listIdx ][ rplIdx ]` igual a 0 especifica que la *i*-ésima entrada en `ref_pic_list_struct( listIdx, rplIdx, ltrpFlag )` tiene un valor menor que 0. Cuando no está presente, el valor de `strp_entries_sign_flag[ listIdx ][ rplIdx ][ i ]` se infiere como igual al valor de `strp_entries_sign_flag[ listIdx ][ rplIdx ]`.

El `delta_poc_st[ listIdx ][ rplIdx ][ i ]`, cuando la *i*-ésima entrada es la primera entrada de STRP en la estructura sintáctica `ref_pic_list_struct ( rplIdx, ltrpFlag )`, especifica la diferencia entre los valores de recuento de orden de imágenes de la imagen actual y de la imagen a que se hace referencia por la *i*-ésima entrada, o, cuando la *i*-ésima entrada es una entrada de STRP pero no la primera entrada de STRP en la estructura sintáctica `ref_pic_list_struct ( rplIdx, ltrpFlag )`, especifica la diferencia entre los valores de recuento de orden de imagen de las imágenes a que se hace referencia por la *i*-ésima entrada y por la entrada de STRP previa en la estructura sintáctica `ref_pic_list_struct ( listIdx, rplIdx, ltrpFlag )`. El valor de `delta_poc_st[ listIdx ][ rplIdx ][ i ]` será del orden de  $-215$  a  $215 - 1$ , inclusive. `poc_lsb_lt[ listIdx ][ rplIdx ][ i ]` especifica el valor del módulo de recuento de orden de imagen `MaxLtPicOrderCntLsb` de la imagen a que se hace referencia por la *i*-ésima entrada en la estructura sintáctica `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. La disposición `DeltaPocSt[ listIdx ][ rplIdx ]` es derivada de la siguiente manera:

```

para( i = 0; i < NumEntriesInList[ listIdx ][ rplIdx ]; i++ ) {
    si( !lt_ref_pic_flag[ i ][ RplIdx[ i ] ][ j ] ) {
        DeltaPocSt[ listIdx ][ rplIdx ][ i ] = (strp_entry_sign_flag[ listIdx ][ rplIdx ][ i ] ) ?
45         delta_poc_st[ listIdx ][ rplIdx ][ i ] : 0 - delta_poc_st[ listIdx ][ rplIdx ][ i ]
    }
}

```

El `model_ref_rpl_idx_delta_minus1[ listIdx ][ rplIdx ]` plus 1 especifica la diferencia entre el valor de `rplIdx` y el índice de la referencia `ref_pic_list_struct ( listIdx, rplIdx, ltrpFlag )`. Cuando `rpl_model[ listIdx ][ rplIdx ]` es igual a 1, la variable `RefRplIdx` es derivada de la siguiente manera:

`RefRplIdx = rplIdx - ( model_ref_rpl_idx_delta_minus1[ listIdx ][ rplIdx ] + 1 )`

El `strp_offset_val_minus1[ listIdx ][ rplIdx ]` plus 1 especifica el valor que ha de ser sustraído de cada entrada de STRP en la referencia `ref_pic_list_struct ( listIdx, rplIdx, ltrpFlag )` para calcular el valor de `delta_poc_st[ listIdx ][ rplIdx ][ i ]` de `ref_pic_list_struct ( listIdx, rplIdx, ltrpFlag )` actual. El `ref_entry_used_flag[ listIdx ][ rplIdx ][ i ]` igual a 1 especifica que la *i*-ésima entrada en `ref_pic_list_struct ( 0, RefRplIdx, ltrpFlag )` es utilizada como una entrada en `ref_pic_list_struct ( listIdx, rplIdx, ltrpFlag )`. `ref_entry_used_Flag[ listIdx ][ rplIdx ]` igual a 0 especifica que la *i*-ésima entrada en `ref_pic_list_struct ( 0, RefRplIdx, ltrpFlag )` no es utilizada como una entrada en `ref_pic_list_struct ( listIdx, rplIdx, ltrpFlag )`.

Cuando `rpl_mode[ listIdx ][ rplsIdx ]` es igual a 1, se aplica lo siguiente para la inferencia de los valores de los elementos de sintaxis `lt_ref_pic_flag[ listIdx ][ rplsIdx ][ i ]`, y cuando `lt_ref_pic_flag[ listIdx ][ rplsIdx ][ i ]` es igual a 1, `poc_lsb_lt[ listIdx ][ rplsIdx ][ i ]`, y para derivación de la variable `DeltaPocSt[ listIdx ][ rplsIdx ][ i ]` (cuando `lt_ref_pic_flag[ listIdx ][ rplsIdx ][ i ]` es igual a 0) y la variable `NumEntriesInList[ listIdx ][ rplsIdx ]`:

```

5 currEntryIdx = 0
  para( i = 0; i < NumEntriesInList[ 0 ][ RefRplsIdx ]; i++ ) {
    si( ref_entry_used_flag[ listIdx ][ rplsIdx ][ i ] ) {
      lt_ref_pic_flag[ listIdx ][ rplsIdx ][ currEntryIdx ] = lt_ref_pic_flag[ 0 ][ RefRplsIdx ][ i ]
      si( !lt_ref_pic_flag[ 0 ][ RefRplsIdx ][ i ] )
10      DeltaPocSt[ listIdx ][ rplsIdx ][ currEntryIdx ] = DeltaPocSt[ 0 ][ RefRplsIdx ][ i ]
      además
        poc_lsb_lt[ listIdx ][ rplsIdx ][ currEntryIdx ] = poc_lsb_lt[ 0 ][ RefRplsIdx ][ i ]
        currEntryIdx++
    }
15 }
  NumEntriesInList[ listIdx ][ rplsIdx ] = currEntryIdx

```

El `mode2_ref_rpl_idx[ listIdx ][ rplsIdx ]` especifica el índice de la referencia `ref_pic_list_struct( listIdx, rplsIdx, ltrpFlag )`. El elemento de sintaxis `mode2_ref_rpl_idx[ listIdx ][ rplsIdx ]` es representado por `Ceil( Log2( num_ref_pic_lists_in_sps[ 0 ] ) )` bits. Cuando `rpl_mode[ listIdx ][ rplsIdx ]` es igual a 2, la variable `RefRplIdx` se deriva de la siguiente manera:

```
RefRplIdx = mode2_ref_rpl_idx[ listIdx ][ rplsIdx ]
```

El `num_additional_entries [ listIdx ][ rplsIdx ]` especifica el delta entre `NumEntriesInList[ listIdx ][ rplsIdx ]` y `NumEntriesInList[ 0 ][ RefRplIdx ]`. El `add_lt_ref_pic_flag [ listIdx ][ rplsIdx ][ i ]` es utilizado para la inferencia del valor de `lt_ref_pic_flag[ listIdx ][ rplsIdx ][ NumEntriesInList[ 0 ][ RefRplIdx ] + i ]`. Cuando no está presente, el valor de `add_lt_ref_pic_flag[ listIdx ][ rplsIdx ][ i ]` se infiere para que sea igual a 0. El `add_stp_entry_sign_flag [ listIdx ][ rplsIdx ][ i ]` es utilizado para la inferencia del valor de `stp_entry_sign_flag [ listIdx ][ rplsIdx ][ NumEntriesInList[ 0 ][ RefRplIdx ] + i ]`. Cuando no está presente, el valor de `add_stp_entries_sign_flag[ listIdx ][ rplsIdx ][ i ]` es establecido para que sea igual al valor de `stp_entries_sign_flag[ listIdx ][ rplsIdx ]`.

```

30 El add_delta_poc_st[ listIdx ][ rplsIdx ][ i ] es utilizado para la inferencia del valor de delta_poc_st[ listIdx ][ rplsIdx ][ NumEntriesInList[ 0 ][ RefRplIdx ] + i ]. El valor de add_delta_poc_st[ listIdx ][ rplsIdx ][ i ] será del orden de -215 a 215 - 1, inclusive. El add_poc_lsb_lt[ listIdx ][ rplsIdx ][ i ] es utilizado para la inferencia del valor de poc_lsb_lt[ listIdx ][ rplsIdx ][ NumEntriesInList[ 0 ][ RefRplIdx ] + i ]. La longitud del elemento de sintaxis add_poc_lsb_lt[ listIdx ][ rplsIdx ][ i ] es Log2( MaxLtPicOrderCntLsb ) bits.

```

Cuando `rpl_mode[ listIdx ][ rplsIdx ]` es igual a 2, se aplica lo siguiente para la inferencia de los valores de los elementos de sintaxis `stp_entries_sign_flag[ listIdx ][ rplsIdx ]`, `lt_ref_pic_flag[ listIdx ][ rplsIdx ][ i ]`, y cuando `lt_ref_pic_flag[ listIdx ][ rplsIdx ][ i ]` es igual a 1, `poc_lsb_lt[ listIdx ][ rplsIdx ][ i ]`, y para derivación de la variable `DeltaPocSt[ listIdx ][ rplsIdx ][ i ]` (cuando `lt_ref_pic_flag[ listIdx ][ rplsIdx ][ i ]` es igual a 0) y la variable `NumEntriesInList[ listIdx ][ rplsIdx ]`:

```

40 stp_entries_sign_flag[ listIdx ][ rplsIdx ] = stp_entries_sign_flag[ 0 ][ RefRplsIdx ]
  para( i = 0; i < NumEntriesInList[ 0 ][ RefRplsIdx ]; i++ ) {
    lt_ref_pic_flag[ listIdx ][ rplsIdx ][ i ] = lt_ref_pic_flag[ 0 ][ RefRplsIdx ][ i ]
    si( !lt_ref_pic_flag[ 0 ][ RefRplsIdx ][ i ] )
45      DeltaPocSt[ listIdx ][ rplsIdx ][ i ] = DeltaPocSt[ 0 ][ RefRplsIdx ][ i ]
    además
      poc_lsb_lt[ listIdx ][ rplsIdx ][ i ] = poc_lsb_lt[ 0 ][ RefRplsIdx ][ i ]
  }
  currEntryIdx = NumEntriesInList[ 0 ][ RefRplsIdx ]
  para( i = 0; i < num_additional_entries[ listIdx ][ rplsIdx ]; i++, currEntryIdx++ ) {
50      lt_ref_pic_flag[ listIdx ][ rplsIdx ][ currEntryIdx ] = add_lt_ref_pic_flag[ listIdx ][ rplsIdx ][ i ]
      si( !add_lt_ref_pic_flag[ listIdx ][ rplsIdx ][ i ] )
        DeltaPocSt[ listIdx ][ rplsIdx ][ currEntryIdx ] = ( add_stp_entry_sign_flag[ listIdx ][ rplsIdx ][ i ] ) ?
          add_delta_poc_st[ listIdx ][ rplsIdx ][ i ] : 0 - add_delta_poc_st[ listIdx ][ rplsIdx ][ i ]
      además
55      poc_lsb_lt[ listIdx ][ rplsIdx ][ currEntryIdx ] = add_poc_lsb_lt[ listIdx ][ rplsIdx ][ i ]
  }

```

NumEntriesInList[ listIdx ][ rplIdx ] = currEntryIdx

El mode3\_ref\_rpl\_idx[ listIdx ][ rplIdx ] especifica el índice de la referencia ref\_pic\_list\_struct ( listIdx, rplIdx, ltrpFlag ). El elemento de sintaxis mode3\_ref\_rpl\_idx[ listIdx ][ rplIdx ] es representado por Ceil( Log2( num\_ref\_pic\_lists\_in\_sps[ 0 ] ) ) bits. Cuando rpl\_mode[ listIdx ][ rplIdx ] es igual a 3, la variable RefRplIdx se deriva de la siguiente manera:

RefRplIdx = mode3\_ref\_rpl\_idx[ listIdx ][ rplIdx ]

Cuando rpl\_mode[ listIdx ][ rplIdx ] es igual a 3, se aplica lo siguiente para la inferencia de los valores de los elementos de sintaxis lt\_ref\_pic\_flag[ listIdx ][ rplIdx ][ i ], y cuando lt\_ref\_pic\_flag[ listIdx ][ rplIdx ][ i ] es igual a 1, poc\_lsb\_lt[ listIdx ][ rplIdx ][ i ], y para derivación de la variable DeltaPocSt[ listIdx ][ rplIdx ][ i ] (cuando lt\_ref\_pic\_flag[ listIdx ][ rplIdx ][ i ] es igual a 0) y la variable NumEntriesInList[ listIdx ][ rplIdx ]:

```
para( i = 0; i < NumEntriesInList[ 0 ][ RefRplIdx ]; i++) {
    lt_ref_pic_flag[ listIdx ][ rplIdx ][ i ] = lt_ref_pic_flag[ 0 ][ RefRplIdx ][ i ]
    si( !lt_ref_pic_flag[ 0 ][ RefRplIdx ][ i ] )
        DeltaPocSt[ listIdx ][ rplIdx ][ i ] = 0 - DeltaPocSt[ 0 ][ RefRplIdx ][ i ]
    además
        poc_lsb_lt[ listIdx ][ rplIdx ][ i ] = poc_lsb_lt[ 0 ][ RefRplIdx ][ i ]
    NumEntriesInList[ listIdx ][ rplIdx ] = NumEntriesInList[ 0 ][ RefRplIdx ]
}
```

Se proporciona un proceso de decodificación general.

El proceso de decodificación opera de la siguiente manera para la imagen actual CurrPic: La decodificación de unidades NAL se especifica a continuación. Los procesos descritos a continuación especifican los siguientes procesos de decodificación utilizando elementos de sintaxis en la capa de cabecera de segmento y anteriormente: se derivan variables y funciones relacionadas con el recuento de orden de imágenes. A esto se recurre únicamente para el primer segmento de una imagen. Al inicio del proceso de decodificación para cada segmento de una imagen no-IRAP, se recurre al proceso de decodificación para construcción de listas de imágenes de referencia para derivación de la lista 0 de imágenes de referencia (RefPicList[ 0 ]) y la lista 1 de imágenes de referencia (RefPicList[ 1 ]). Se recurre al proceso de decodificación para marcación de imágenes de referencia, en donde imágenes de referencia pueden ser marcadas como “no utilizada para referencia” o “utilizada para referencia a largo plazo.” A esto se recurre únicamente para el primer segmento de una imagen. Se recurre a los procesos de decodificación para codificar unidades de árbol, escalado, transformada, filtración en-bucle, etc. Después que todos los segmentos de la imagen actual han sido decodificados, la imagen decodificada actual es marcada como “utilizada para referencia a corto plazo.”

Se proporciona un proceso de decodificación de unidad NAL.

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

Se proporciona un proceso de decodificación de segmento.

El proceso de decodificación para recuento de orden de imágenes es como sigue.

La salida de este proceso es PicOrderCntVal, el recuento de orden de imágenes de la imagen actual. Los recuentos de orden de imágenes son utilizados para identificar imágenes, para derivar parámetros de movimiento en modo de fusión y predicción de vector de movimiento, y para comprobación de conformidad del decodificador. Cada imagen codificada está asociada con una variable de recuento de orden de imágenes, indicada como PicOrderCntVal. Cuando la imagen actual no es una imagen IRAP, las variables prevPicOrderCntLsb y prevPicOrderCntMsb son derivadas de la siguiente manera: Asumir que prevTid0Pic es la imagen anterior en el orden de decodificación que tiene TemporalId igual a 0. La variable prevPicOrderCntLsb se establece igual a slice\_pic\_order\_cnt\_lsb de prevTid0Pic. La variable prevPicOrderCntMsb se establece igual a PicOrderCntMsb de prevTid0Pic.

La variable PicOrderCntMsb de la imagen actual se deriva de la siguiente manera: Si la imagen actual es una imagen IRAP, PicOrderCntMsb se establece igual a 0. De otra manera, PicOrderCntMsb se deriva de la siguiente manera:

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

además si( (slice\_pic\_order\_cnt\_lsb > prevPicOrderCntLsb) &&

(( slice\_pic\_order\_cnt\_lsb - prevPicOrderCntLsb ) > ( MaxPicOrderCntLsb / 2 ))

PicOrderCntMsb = prevPicOrderCntMsb - MaxPicOrderCntLsb

además

5 PicOrderCntMsb = prevPicOrderCntMsb

PicOrderCntVal se deriva como sigue: PicOrderCntVal = PicOrderCntMsb + slice\_pic\_order\_cnt\_lsb.

Todas las imágenes IRAP tendrán PicOrderCntVal igual a 0 dado que slice\_pic\_order\_cnt\_lsb se infiere para que sea 0 para imágenes IRAP y prevPicOrderCntLsb y prevPicOrderCntMsb se establecen iguales a 0. El valor de PicOrderCntVal será del orden de  $-231$  a  $231 - 1$ , inclusive. En un CVS, los valores PicOrderCntVal para cualesquiera dos imágenes codificadas no serán los mismos.

10

En cualquier momento durante el proceso de decodificación, los valores de PicOrderCntVal & ( MaxLtPicOrderCntLsb - 1 ) para cualesquiera dos imágenes de referencia en la DPB no serán los mismos. La función PicOrderCnt(picX) se especifica como sigue: 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). La corriente de bits no contendrá datos que den como resultado valores de DiffPicOrderCnt(picA, picB) utilizados en el proceso de decodificación que no estén dentro del intervalo de  $-215$  a  $215 - 1$ , inclusive. Asumir que X es la imagen actual e Y y Z son otras dos imágenes en el mismo CVS, se considera que Y y Z están en la misma dirección de orden de salida de X cuando tanto DiffPicOrderCnt(X, Y) como DiffPicOrderCnt(X, Z) son positivos o ambos son negativos.

20

Se proporciona un proceso de decodificación para la construcción de listas de imágenes de referencia.

Se recurre a este proceso al inicio del proceso de decodificación para cada segmento de una imagen no-IRAP. Imágenes de referencia son abordadas a través de índices de referencia. Un índice de referencia es un índice en una lista de imágenes de referencia. Cuando se decodifica un segmento I, no se utiliza ninguna lista de imágenes de referencia en la decodificación de los datos de segmento. Cuando se decodifica un segmento P, solamente se utiliza la lista 0 de imágenes de referencia (es decir, RefPicList[0]), en la decodificación de los datos de segmento. Cuando se decodifica un segmento B, tanto la lista 0 de imágenes de referencia como la lista 1 de imágenes de referencia (es decir, RefPicList[1]) son utilizadas en la decodificación de los datos de segmento. Al inicio del proceso de decodificación para cada segmento de una imagen no-IRAP, las listas de imágenes de referencia RefPicList[0] y RefPicList[1] son derivadas. Las listas de imágenes de referencia son utilizadas en la marcación de imágenes de referencia o en la decodificación de los datos de segmento. Para un segmento I de una imagen no-IRAP que no es el primer segmento de la imagen, RefPicList[0] y RefPicList[1] pueden ser derivadas con el propósito de comprobación de conformidad de corriente de bits, pero su derivación no es necesaria para decodificar la imagen o imágenes actuales después de la imagen actual en el orden de decodificación. Para un segmento P que no es el primer segmento de una imagen, RefPicList[1] puede ser derivada con el propósito de comprobación de conformidad de corriente de bits, pero su derivación no es necesaria para decodificar la imagen o imágenes actuales después de la imagen actual en el orden de decodificación. Las listas de imágenes de referencia RefPicList[0] y RePicList[1] son construidas de la siguiente manera:

25

30

35

40

para( i = 0; i < 2; i++ ) {

    si( ref\_pic\_list\_sps\_flag[ i ] )  
        RplIdx[ i ] = ref\_pic\_list\_idx[ i ]

    además

        RplIdx[ i ] = num\_ref\_pic\_lists\_in\_sps[ i ]

45

    para( j = 0, pocBase = PicOrdenCntVal; j < NumEntriesInList[ i ][ RplIdx[ i ] ]; j++ ) {

        si( !lt\_ref\_pic\_flag[ i ][ RplIdx[ i ] ][ j ] ) {  
            RefPicPocList[ i ][ j ] = pocBase - DeltaPocSt[ i ][ RplIdx[ i ] ][ j ]

        si( hay una imagen de referencia picA en la DPB con PicOrderCntVal igual a RefPicPocList[ i ][ j ] )

50

            RefPicList[ i ][ j ] = picA

        además

            RefPicList[ i ][ j ] = "no imagen de referencia"

        pocBase = RefPicPocList[ i ][ j ]

    } además {

55

        si( hay una referencia picA en la DPB con PicOrderCntVal & ( MaxLtPicOrderCntLsb - 1 )  
            igual a poc\_lsb\_lt[ i ][ RplIdx[ i ] ][ j ] )

            RefPicList[ i ][ j ] = picA

```

    además
      RefPicList[ i ][ j ] = "no imagen de referencia"
    }
  }
5 }

```

Para cada  $i$  igual a 0 o 1, se aplica lo siguiente: Las primeras entradas  $\text{NumRefIdxActive}[i]$  en  $\text{RefPicList}[i]$  son denominadas como las entradas activas en  $\text{RefPicList}[i]$ , y las otras entradas en  $\text{RefPicList}[i]$  son denominadas como las entradas inactivas en  $\text{RefPicList}[i]$ . Cada entrada en  $\text{RefPicList}[i][j]$  para  $j$  en el intervalo de 0 a  $\text{NumEntriesInList}[i][\text{RplIdx}[i]] - 1$ , inclusive, se denomina como una entrada de STRP si  $\text{lt\_ref\_pic\_flag}[i][\text{RplIdx}[i]][j]$  es igual a 0, y como una entrada de LTRP en caso contrario. Es posible que una imagen particular sea hecha referencia tanto por una entrada en  $\text{RefPicList}[0]$  como por una entrada en  $\text{RefPicList}[1]$ . También es posible que una imagen particular sea hecha referencia por más de una entrada en  $\text{RefPicList}[0]$  o por más de una entrada en  $\text{RefPicList}[1]$ . Las entradas activas en  $\text{RefPicList}[0]$  y las entradas activas en  $\text{RefPicList}[1]$  colectivamente se refieren a todas las imágenes de referencia que pueden ser utilizadas para inter-predicción de la imagen actual y una o más imágenes que siguen a la imagen actual en el orden de decodificación. Las entradas inactivas en  $\text{RefPicList}[0]$  y las entradas inactivas en  $\text{RefPicList}[1]$  colectivamente se refieren a todas las imágenes de referencia que no son utilizadas para inter-predicción de la imagen actual pero que pueden ser utilizadas en inter-predicción para una o más imágenes que siguen a la imagen actual en el orden de decodificación. Puede haber una o más entradas en  $\text{RefPicList}[0]$  o  $\text{RefPicList}[1]$  que sean iguales a "no imagen de referencia" debido a que las imágenes correspondientes no están presentes en la DPB. Cada entrada inactiva en  $\text{RefPicList}[0]$  o  $\text{RefPicList}[1]$  que es igual a "no imagen de referencia" debería ser ignorada. Una pérdida de imagen no intencionada debería ser inferida para cada entrada activa en  $\text{RefPicList}[0]$  o  $\text{RefPicList}[1]$  que sea igual a "no imagen de referencia."

Un requerimiento de la conformidad de corriente de bits es 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 referida por cada entrada activa en  $\text{RefPicList}[0]$  o  $\text{RefPicList}[1]$  estará presente en la DPB y tendrá  $\text{TemporalId}$  menor que o igual al de la imagen actual. Opcionalmente se puede especificar la siguiente restricción: El índice de entrada de cualquier entrada inactiva en  $\text{RefPicList}[0]$  o  $\text{RefPicList}[1]$  no se utilizará como un índice de referencia para la decodificación de la imagen actual. Opcionalmente se puede especificar 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 de STRP en  $\text{RefPicList}[0]$  o  $\text{RefPicList}[1]$  de un segmento de una imagen y una entrada de LTRP en  $\text{RefPicList}[0]$  o  $\text{RefPicList}[1]$  del mismo segmento o de un segmento diferente de la misma imagen no se referirá a la misma imagen. La propia imagen actual no será referida por ninguna entrada en  $\text{RefPicList}[0]$  o  $\text{RefPicList}[1]$ . No habrá una entrada de LTRP en  $\text{RefPicList}[0]$  o  $\text{RefPicList}[1]$  para la que la diferencia entre  $\text{PicOrderCntVal}$  de la imagen actual y  $\text{PicOrderCntVal}$  de la imagen referida por la entrada sea mayor que o igual a 224. Asumir que  $\text{setOfRefPics}$  sea el conjunto de imágenes únicas referidas por todas las entradas en  $\text{RefPicList}[0]$  y todas las entradas en  $\text{RefPicList}[1]$ . El número de imágenes en  $\text{setOfRefPics}$  será menor que o igual a  $\text{sps\_max\_dec\_pic\_buffering\_minus1}$  y  $\text{setOfRefPics}$  será el mismo para todos los segmentos de una imagen.

Se proporciona un proceso de decodificación para marcación de imagen de referencia.

Se recurre a este proceso una vez por imagen, después de la decodificación de una cabecera de segmento y del proceso de decodificación para la construcción de lista de imágenes de referencia para el segmento, pero antes de la decodificación de los datos de segmento. Este proceso puede dar como resultado que una o más imágenes de referencia en la DPB sean marcadas como "no utilizada para referencia" o "utilizada para referencia a largo plazo." Una imagen decodificada en la DPB puede ser marcada como "no utilizada para referencia," "utilizada para referencia a corto plazo" o "utilizada para referencia a largo plazo," pero solamente una de entre estas tres en cualquier momento determinado durante la operación del proceso de decodificación. La asignación de una de estas marcaciones a una imagen elimina implícitamente otra de estas marcaciones cuando es aplicable. Cuando una imagen es referida como que es marcada como "utilizada para referencia," esto se refiere colectivamente a la imagen que es marcada como "utilizada para referencia a corto plazo" o "utilizada para referencia a largo plazo" (pero no a ambas). Cuando la imagen actual es una imagen IRAP, todas las imágenes de referencia actualmente en la DPB (si las hay) son marcadas como "no utilizada para referencia." Las STRP son identificadas por sus valores  $\text{PicOrderCntVal}$ . Las LTRP son identificadas por los  $\text{LBS Log2}(\text{MaxLtPicOrderCntLsb})$  de sus valores  $\text{PicOrderCntVal}$ . Se aplica lo siguiente: Para cada entrada de LTRP en  $\text{RefPicList}[0]$  o  $\text{RefPicList}[1]$ , cuando la imagen referida es una STRP, la imagen es marcada como "utilizada para referencia a largo plazo". Cada imagen de referencia en la DPB que no es referida por ninguna entrada en  $\text{RefPicList}[0]$  o  $\text{RefPicList}[1]$  es marcada como "no utilizada para referencia".

La figura 8 es un diagrama esquemático de un dispositivo 800 de codificación de video (por ejemplo, un codificador 20 de video o un decodificador 30 de video) según una realización de la descripción. El dispositivo

800 de codificación de video es adecuado para implementar las realizaciones descritas como se describe en la presente memoria. El dispositivo 800 de codificación de video comprende puertos 810 de entrada y unidades receptoras (Rx) 820 para recibir datos; un procesador, unidad lógica, o unidad de procesamiento central (CPU) 830 para procesar los datos; unidades transmisoras (Tx) 840 y puertos 850 de salida para transmitir los datos; y una memoria 860 para almacenar los datos. El dispositivo 800 de codificación de video también puede comprender componentes óptico-a-eléctrico (OE) y componentes eléctrico-a-óptico (EO) acoplados a los puertos 810 de entrada, las unidades receptoras 820, las unidades transmisoras 840, y los puertos 850 de salida para salida o entrada de señales ópticas o eléctricas.

El procesador 830 es implementado por hardware y software. El procesador 830 puede ser implementado como uno o más chips CPU, núcleos (por ejemplo, como un procesador multi-núcleo), agrupaciones de puertas programables en campo (FPGA), circuitos integrados de aplicación específica (ASIC), y procesadores de señal digital (DSP). El procesador 830 está en comunicación con los puertos 810 de entrada, unidades receptoras 820, unidades transmisoras 840, puertos 850 de salida, y memoria 860. El procesador 830 comprende un módulo 870 de codificación. El módulo 870 de codificación implementa las realizaciones descritas anteriormente. Por ejemplo, el módulo 870 de codificación implementa, procesa, prepara, o proporciona las distintas funciones de conexión en red. La inclusión del módulo 870 de codificación proporciona por ello una mejora sustancial de la funcionalidad del dispositivo 800 de codificación de video y efectúa una transformación del dispositivo 800 de codificación de video a un estado diferente. Alternativamente, el módulo 870 de codificación es implementado como instrucciones almacenadas en la memoria 860 y es ejecutado por el procesador 830.

El dispositivo 800 de codificación de video también puede incluir dispositivos 880 de entrada y/o salida (I/O) para comunicar datos hacia y desde un usuario. Los dispositivos 880 de I/O pueden incluir dispositivos de salida tales como un dispositivo de visualización para mostrar datos de video, altavoces para emitir datos de audio, etc. Los dispositivos 880 de I/O también pueden incluir dispositivos de entrada, tales como un teclado, ratón, bola de seguimiento, etc., y/o interfaces correspondientes para interactuar con dichos dispositivos de salida.

La memoria 860 comprende uno o más discos, unidades de cinta, y unidades de estado sólido y se puede utilizar como un dispositivo de almacenamiento de datos de sobre-flujo, para almacenar programas cuando dichos programas son seleccionados para ejecución, y para almacenar instrucciones y datos que son leídos durante la ejecución del programa. La memoria 860 puede ser volátil y/o no volátil y puede ser memoria de solo lectura (ROM), memoria de acceso aleatorio (RAM), memoria de contenido direccionable ternario (TCAM), y/o memoria de acceso aleatorio estática (SRAM).

La figura 9 es un diagrama esquemático de una realización de un medio 900 para codificación. En la realización, el medio 900 para codificación es implementado en un dispositivo 902 de codificación de video (por ejemplo, un codificador 20 de video o un decodificador 30 de video). El dispositivo 902 de codificación de video incluye medios 901 de recepción. Los medios 901 de recepción están configurados para recibir una imagen a codificar o para recibir una corriente de bits a decodificar. El dispositivo 902 de codificación de video incluye medios 907 de transmisión acoplados a los medios 901 de recepción. Los medios 907 de transmisión están configurados para transmitir la corriente de bits a un decodificador o para transmitir una imagen decodificada a un medio de visualización (por ejemplo, uno de los dispositivos 880 de I/O).

El dispositivo 902 de codificación de video incluye un medios 903 de almacenamiento. El medio 903 de almacenamiento está acoplado al menos a uno de los medios receptores 901 o de los medios 907 de transmisión. El medio 903 de almacenamiento está configurado para almacenar instrucciones. El dispositivo 902 de codificación de video también incluye medios 905 de procesamiento. El medio 905 de procesamiento está acoplado al medio 903 de almacenamiento. El medio 905 de procesamiento está configurado para ejecutar las instrucciones almacenadas en el medio 903 de almacenamiento para ejecutar los métodos descritos en la presente memoria.

## REIVINDICACIONES

1. Un método para decodificar una corriente de bits de video codificada implementado por un decodificador de video, que comprende:

analizar, a partir de la corriente de bits de video codificada, un indicador;

5 analizar, a partir de la corriente de bits de video codificada, un `num_ref_pic_lists_in_sps[ 0 ]` y estructuras sintácticas `ref_pic_list_struct(0, rplsIdx, ltrpFlag)` para una primera lista de imágenes de referencia; en donde el `num_ref_pic_lists_in_sps[ 0 ]` especifica un número de las estructuras sintácticas `ref_pic_list_struct(0, rplsIdx, ltrpFlag)` incluidas en un conjunto de parámetros de secuencia, SPS, de la corriente bits de video codificada; en donde `num_ref_pic_lists_in_sps[ 0 ]` es del orden de 0 a 64, el `ltrpFlag` en la `ref_pic_list_struct(0, rplsIdx, ltrpFlag)` es igual a 0 o 1;

10 cuando el indicador tiene un primer valor, determinar que el `num_ref_pic_lists_in_sps[ 1 ]` y las estructuras sintácticas `ref_pic_list_struct(1, rplsIdx, ltrpFlag)` para una segunda lista de imágenes de referencia no están presentes en la corriente de bits de video codificada, estableciendo un valor de `num_ref_pic_lists_in_sps[ 1 ]` para que sea igual a un valor del `num_ref_pic_lists_in_sps[ 0 ]`, y en donde los valores de los elementos de sintaxis en cada una de las estructuras sintácticas de `ref_pic_list_struct(1, rplsIdx, ltrpFlag)` para la segunda lista de imágenes de referencia se infieren para que sean respectivamente iguales a los valores de los elementos de sintaxis en cada una de las estructuras sintácticas de `ref_pic_list_struct(0, rplsIdx, ltrpFlag)` para la primera lista de imágenes de referencia; en donde el `num_ref_pic_lists_in_sps[ 1 ]` especifica una serie de estructuras sintácticas de `ref_pic_list_struct(1, rplsIdx, ltrpFlag)` en el SPS; en donde `num_ref_pic_lists_in_sps[ 1 ]` es del orden de 0 a 64, el `ltrpFlag` en el `ref_pic_list_struct(1, rplsIdx, ltrpFlag)` es igual a 0 o 1;

15 en donde `ltrpFlag` igual a 0 especifica que la imagen de referencia a largo plazo nc, LTRP, se utiliza para la inter-predicción de cualquier imagen codificada en una secuencia de video codificada, CVS, y `ltrpFlag` igual a 1 especifica que los LTRP pueden utilizarse para inter-predicción de una o más imágenes codificadas en el CVS;

20 en donde para `ref_pic_list_struct(0, rplsIdx, ltrpFlag)`, `rplsIdx` está entre 0 y `num_ref_pic_lists_in_sps[ 0 ] - 1`; para `ref_pic_list_struct(1, rplsIdx, ltrpFlag)`, `rplsIdx` está entre 0 y `num_ref_pic_lists_in_sps[ 1 ] - 1`;

30 cuando el indicador tiene un segundo valor, determinar que el `num_ref_pic_lists_in_sps[ 1 ]` y las estructuras sintácticas `ref_pic_list_struct(1, rplsIdx, ltrpFlag)` para la segunda lista de imágenes de referencia están presentes en la corriente de bits de video codificada, y analizar, a partir de la corriente de bits de video codificada, el `num_ref_pic_lists_in_sps[ 1 ]` y las estructuras sintácticas `ref_pic_list_struct(1, rplsIdx, ltrpFlag)` para la segunda lista de imágenes de referencia;

35 generar la primera lista de imágenes de referencia utilizando una de las estructuras sintácticas `ref_pic_list_struct(0, rplsIdx, ltrpFlag)` para la primera lista de imágenes de referencia a partir de la corriente de bits de video codificada;

generar la segunda lista de imágenes de referencia utilizando una de las estructuras sintácticas `ref_pic_list_struct(1, rplsIdx, ltrpFlag)` para la segunda lista de imágenes de referencia; y

realizar inter-predicción basándose en la primera lista de imágenes de referencia o/y la segunda lista de imágenes de referencia para generar un bloque reconstruido.

40 2. El método según la reivindicación 1, en donde el indicador está designado `rpl1_copy_from_rpl0_flag`.

3. El método según cualquiera de las reivindicaciones 1 a 2, en donde el indicador está incluido en el SPS.

4. El método según cualquiera de las reivindicaciones 1 a 3, en donde las estructuras sintácticas `ref_pic_list_struct( 1, rplsIdx, ltrpFlag )` para la segunda lista de imágenes de referencia no están presentes en el SPS cuando el indicador tiene el primer valor.

45 5. El método según cualquiera de las reivindicaciones 1 a 4, en donde las estructuras sintácticas `ref_pic_list_struct( 1, rplsIdx, ltrpFlag )` para la segunda lista de imágenes de referencia están presentes en el SPS cuando el indicador tiene el segundo valor.

50 6. El método según cualquiera de las reivindicaciones 1 a 5, en donde el indicador está dispuesto en la corriente de bits de video codificada antes que cualquier estructura sintáctica de imagen de referencia predefinida.

7. Un método para codificar una corriente de bits de video implementado por un codificador de video, que comprende:

determinar si un `num_ref_pic_lists_in_sps[ 1 ]` y las estructuras sintácticas `ref_pic_list_struct( 1, rplIdx, ltrpFlag )` para una segunda lista de imágenes de referencia están codificadas en la corriente de bits de vídeo; en donde el `num_ref_pic_lists_in_sps[ 1 ]` especifica una serie de las estructuras sintácticas `ref_pic_list_struct( 1, rplIdx, ltrpFlag )` en un conjunto de parámetros de secuencia, SPS, de la corriente de bits de vídeo codificada; en donde `num_ref_pic_lists_in_sps[ 1 ]` es del orden de 0 a 64, el `ltrpFlag` en el `ref_pic_list_struct( 1, rplIdx, ltrpFlag )` es igual a 0 o 1;

codificar un indicador con un primer valor en la corriente de bits de vídeo cuando se determina que el `num_ref_pic_lists_in_sps[ 1 ]` y las estructuras sintácticas `ref_pic_list_struct( 1, rplIdx, ltrpFlag )` para una segunda lista de imágenes de referencia no están codificadas en la corriente de bits de vídeo, y el indicador con el primer valor especifica que un valor del `num_ref_pic_lists_in_sps[ 1 ]` es igual a un valor de un `num_ref_pic_lists_in_sps[ 0 ]`, y los valores de los elementos de sintaxis en cada una de las estructuras sintácticas `ref_pic_list_struct( 1, rplIdx, ltrpFlag )`, para la segunda lista de imágenes de referencia se infieren respectivamente para que sean iguales a valores de elementos de sintaxis en cada una de las estructuras sintácticas `ref_pic_list_struct( 0, rplIdx, ltrpFlag )` para una primera lista de imágenes de referencia; en donde el `num_ref_pic_lists_in_sps[ 0 ]` especifica un número de las estructuras sintácticas `ref_pic_list_struct( 0, rplIdx, ltrpFlag )` en el SPS; en donde `num_ref_pic_lists_in_sps[ 0 ]` es del orden de 0 a 64; el `ltrpFlag` en el `ref_pic_list_struct( 0, rplIdx, ltrpFlag )` es igual a 0 o a 1;

en donde `ltrpFlag` igual a 0 especifica que la imagen de referencia a largo plazo `nc`, LTRP, se utiliza para inter-predicción de cualquier imagen codificada en una secuencia de vídeo codificada, CVS, y `ltrpFlag` igual a 1 especifica que los LTRP pueden utilizarse para inter-predicción de una o más imágenes codificadas en el CVS;

en donde para `ref_pic_list_struct( 0, rplIdx, ltrpFlag )`, `rplIdx` está entre 0 y `num_ref_pic_lists_in_sps[ 0 ] - 1`; para `ref_pic_list_struct( 1, rplIdx, ltrpFlag )`, `rplIdx` está entre 0 y `num_ref_pic_lists_in_sps[ 1 ] - 1`;

codificar un indicador con un segundo valor en la corriente de bits de vídeo cuando se determina que el `num_ref_pic_lists_in_sps[ 1 ]` y las estructuras sintácticas `ref_pic_list_struct( 1, rplIdx, ltrpFlag )` para una segunda lista de imágenes de referencia están codificadas en la corriente de bits de vídeo;

codificar el `num_ref_pic_lists_in_sps[ 0 ]` y las estructuras sintácticas `ref_pic_list_struct( 0, rplIdx, ltrpFlag )` para la primera lista de imágenes de referencia en la corriente de bits de vídeo cuando el indicador está codificado con el primer valor;

codificar el `num_ref_pic_lists_in_sps[ 0 ]`, y las estructuras sintácticas `ref_pic_list_struct( 0, rplIdx, ltrpFlag )` para la primera lista de imágenes de referencia, el `num_ref_pic_lists_in_sps[ 1 ]` y las estructuras sintácticas `ref_pic_list_struct( 1, rplIdx, ltrpFlag )` para la segunda lista de imágenes de referencia en la corriente de bits de vídeo cuando el indicador está codificado con el segundo valor.

8. El método según con la reivindicación 7, en donde el método comprende además transmitir la corriente de bits de vídeo hacia el decodificador de vídeo.

9. El método según cualquiera de las reivindicaciones 7 a 8, en donde el indicador está codificado en el SPS.

10. El método según cualquiera de las reivindicaciones 7 a 9, en donde las estructuras sintácticas `ref_pic_list_struct( 1, rplIdx, ltrpFlag )` de la segunda lista de imágenes de referencia no están presentes en el SPS cuando el indicador tiene el primer valor.

11. Una corriente de bits codificada, la corriente de bits comprende un `num_ref_pic_lists_in_sps[ 0 ]`, y estructuras sintácticas `ref_pic_list_struct( 0, rplIdx, ltrpFlag )` para una primera lista de imágenes de referencia y un indicador para especificar si un `num_ref_pic_lists_in_sps[ 1 ]` y estructuras sintácticas `ref_pic_list_struct( 1, rplIdx, ltrpFlag )` para una segunda lista de imágenes de referencia están codificadas en la corriente de bits; en donde el `num_ref_pic_lists_in_sps[ 0 ]` especifica un número de las estructuras sintácticas `ref_pic_list_struct( 0, rplIdx, ltrpFlag )`, en un conjunto de parámetros de secuencia, SPS, de la corriente de bits y el `num_ref_pic_lists_in_sps[ 1 ]` especifica un número de la estructura sintáctica `ref_pic_list_struct( 1, rplIdx, ltrpFlag )` en el SPS; en donde `num_ref_pic_lists_in_sps[ 0 ]` es del orden de 0 a 64, el `ltrpFlag` en la `ref_pic_list_struct( 0, rplIdx, ltrpFlag )` es igual a 0 o 1; y en donde `num_ref_pic_lists_in_sps[ 1 ]` es del orden de 0 a 64, el `ltrpFlag` en la `ref_pic_list_struct( 1, rplIdx, ltrpFlag )` es igual a 0 o 1;

en donde `ltrpFlag` igual a 0 especifica que la imagen de referencia a largo plazo `nc`, LTRP, se utiliza para la inter-predicción de cualquier imagen codificada en una secuencia de vídeo codificada, CVS, y `ltrpFlag` igual a 1 especifica que los LTRP pueden utilizarse para la inter-predicción de una o más imágenes codificadas en el CVS;

en donde para `ref_pic_list_struct( 0, rplIdx, ltrpFlag )`, `rplIdx` está entre 0 y `num_ref_pic_lists_in_sps[ 0 ] - 1`; para `ref_pic_list_struct( 1, rplIdx, ltrpFlag )`, `rplIdx` está entre 0 y `num_ref_pic_lists_in_sps[ 1 ] - 1`;

5 en donde el indicador con un primer valor especifica el `num_ref_pic_lists_in_sps[ 1 ]` y las estructuras sintácticas `ref_pic_list_struct( 1, rplsIdx, ltrpFlag )` para la segunda lista de imágenes de referencia no están presentes en la corriente de bits, y un valor de `num_ref_pic_lists_in_sps[ 1 ]` es igual a un valor del `num_ref_pic_lists_in_sps[ 0 ]`, y los valores de los elementos de sintaxis en cada una de las estructuras sintácticas de `ref_pic_list_struct( 1, rplsIdx, ltrpFlag )` para la segunda lista de imágenes de referencia se infieren respectivamente para ser iguales a los valores de los elementos de sintaxis en cada una de las estructuras sintácticas `ref_pic_list_struct( 0, rplsIdx, ltrpFlag )` para la primera lista de imágenes de referencia;

10 en donde el indicador con un segundo valor especifica que el `num_ref_pic_lists_in_sps[ 1 ]` y las estructuras sintácticas de `ref_pic_list_struct( 1, rplsIdx, ltrpFlag )` para la segunda lista de imágenes de referencia están presentes en la corriente bits.

12. Un codificador que comprende circuitos de procesamiento para llevar a la práctica el método según cualquiera de las reivindicaciones 7 a 10.

13. Un decodificador que comprende circuitos de procesamiento para llevar a la práctica el método según cualquiera de las reivindicaciones 1 a 6.

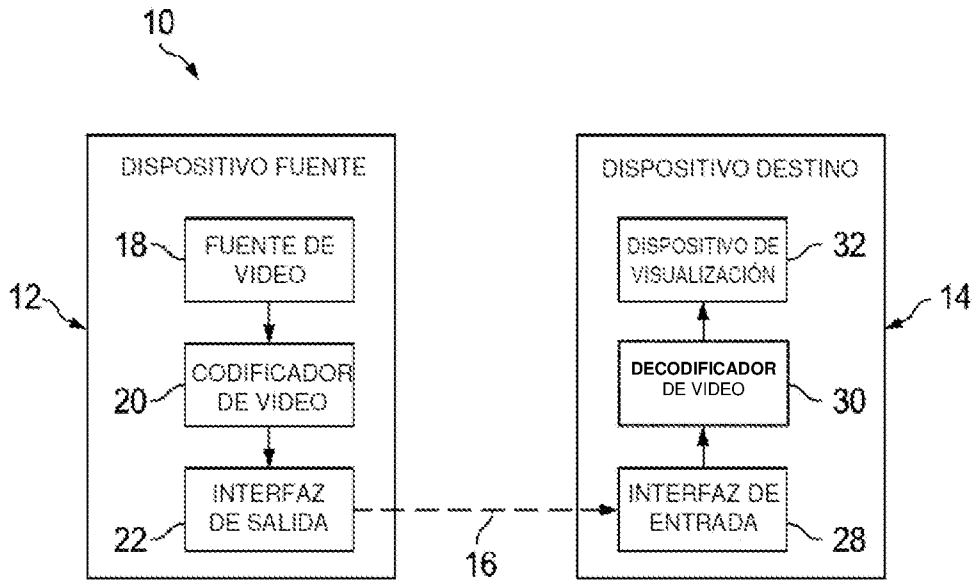


FIGURA 1

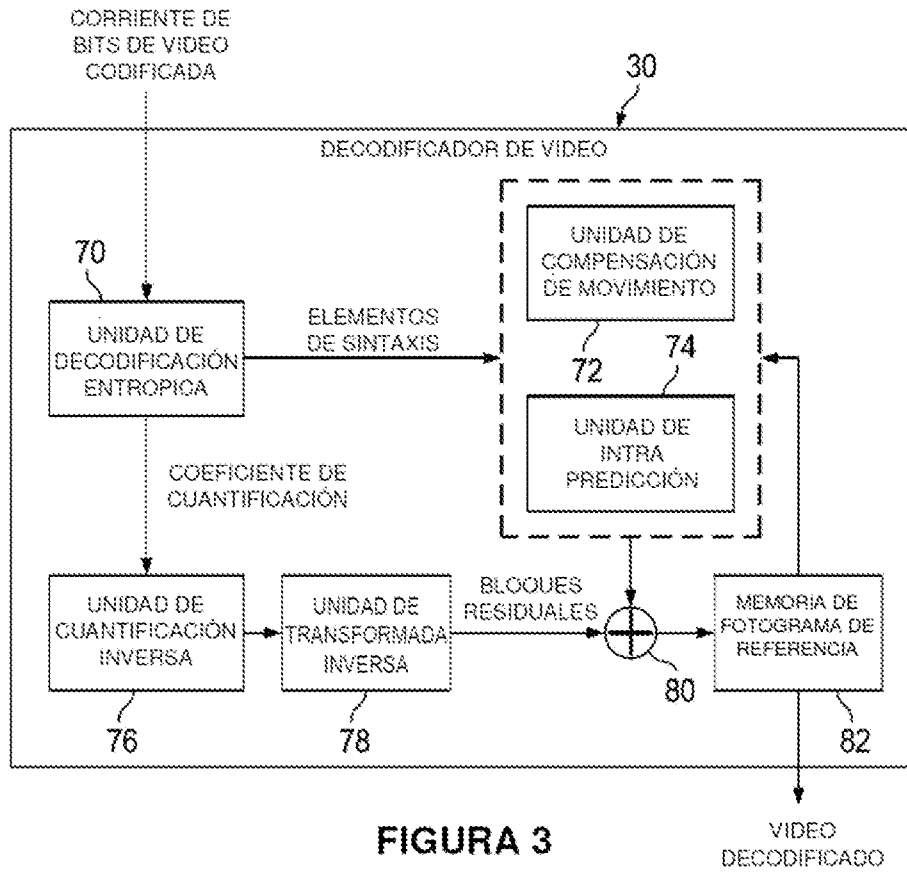


FIGURA 3

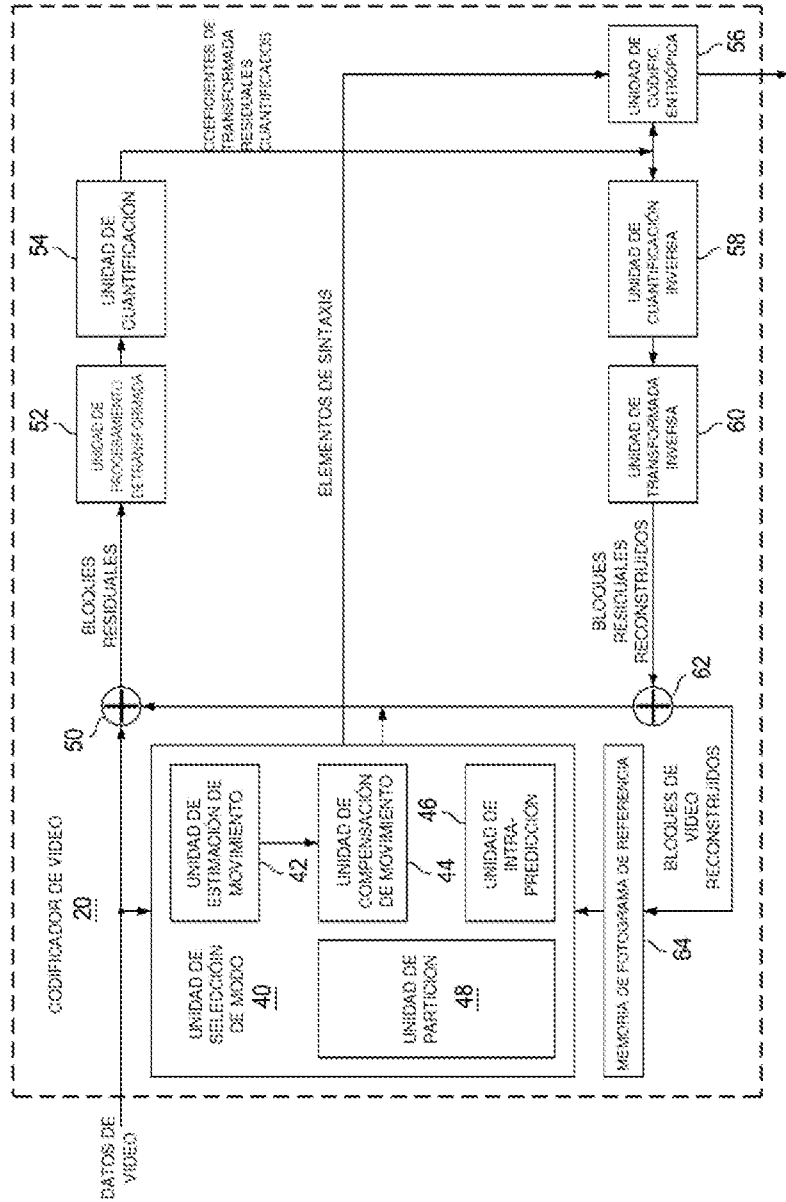


FIGURA 2

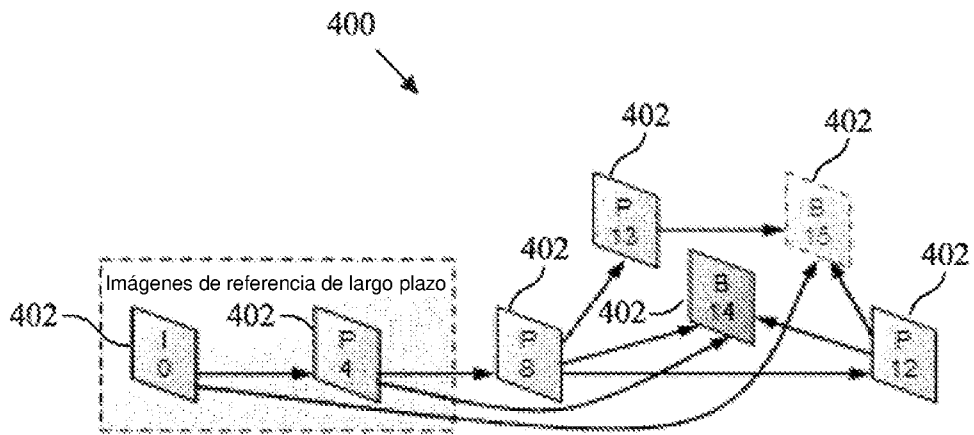
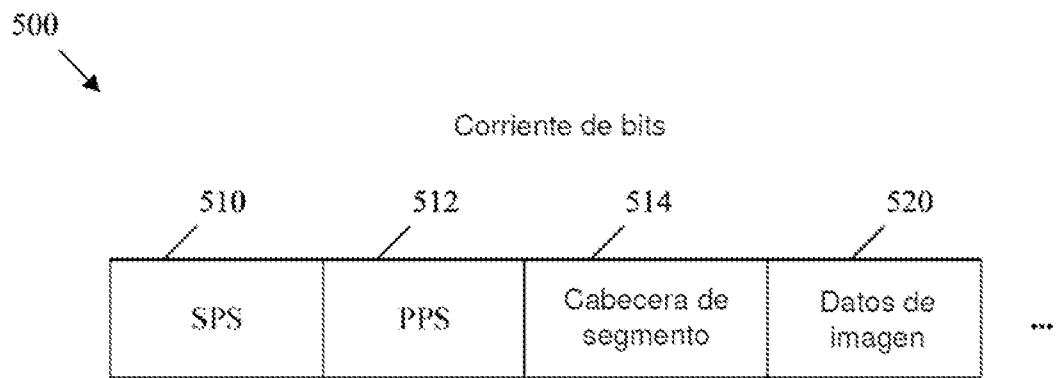


FIGURA 4



**FIGURA 5**

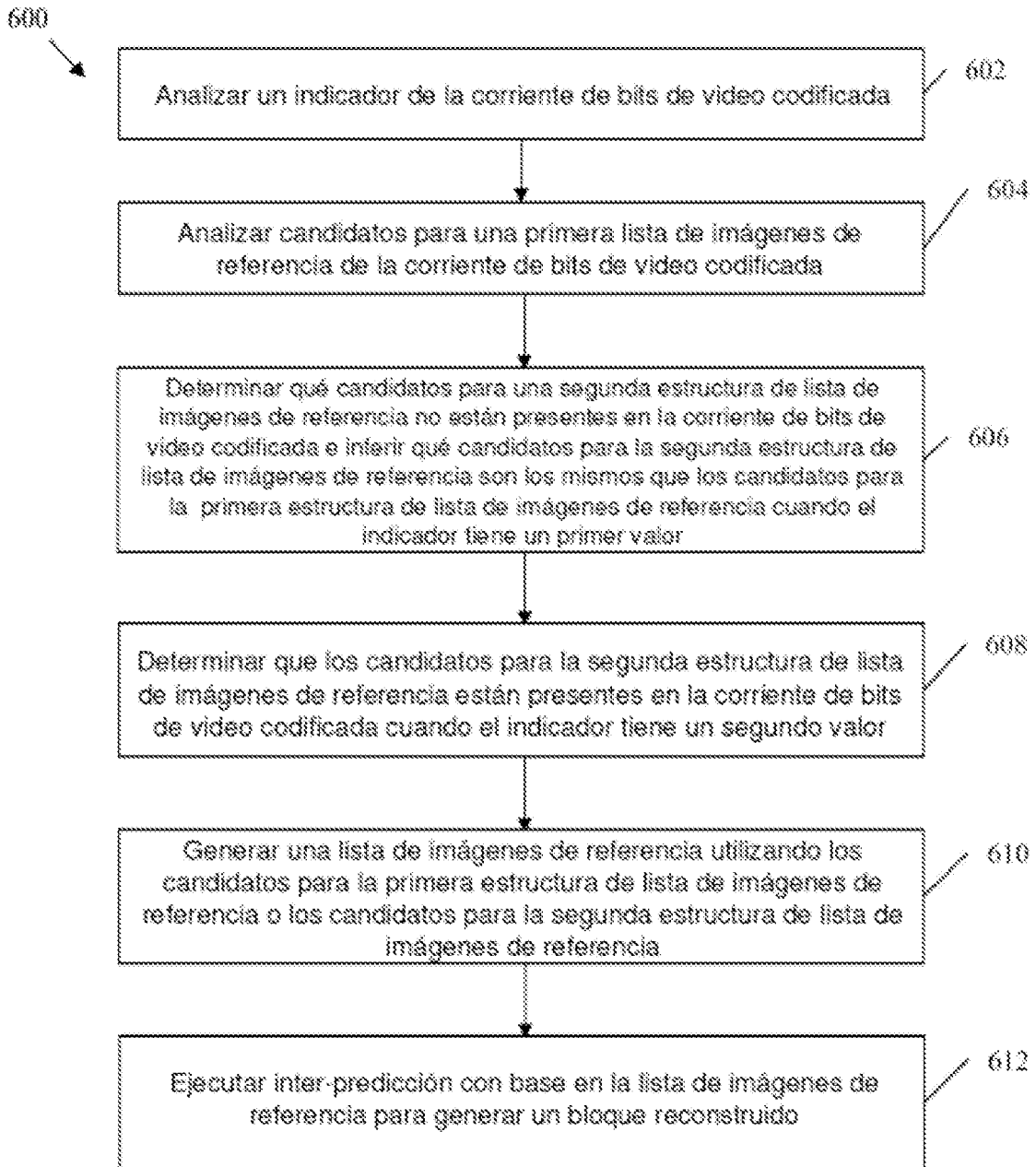


FIGURA 6

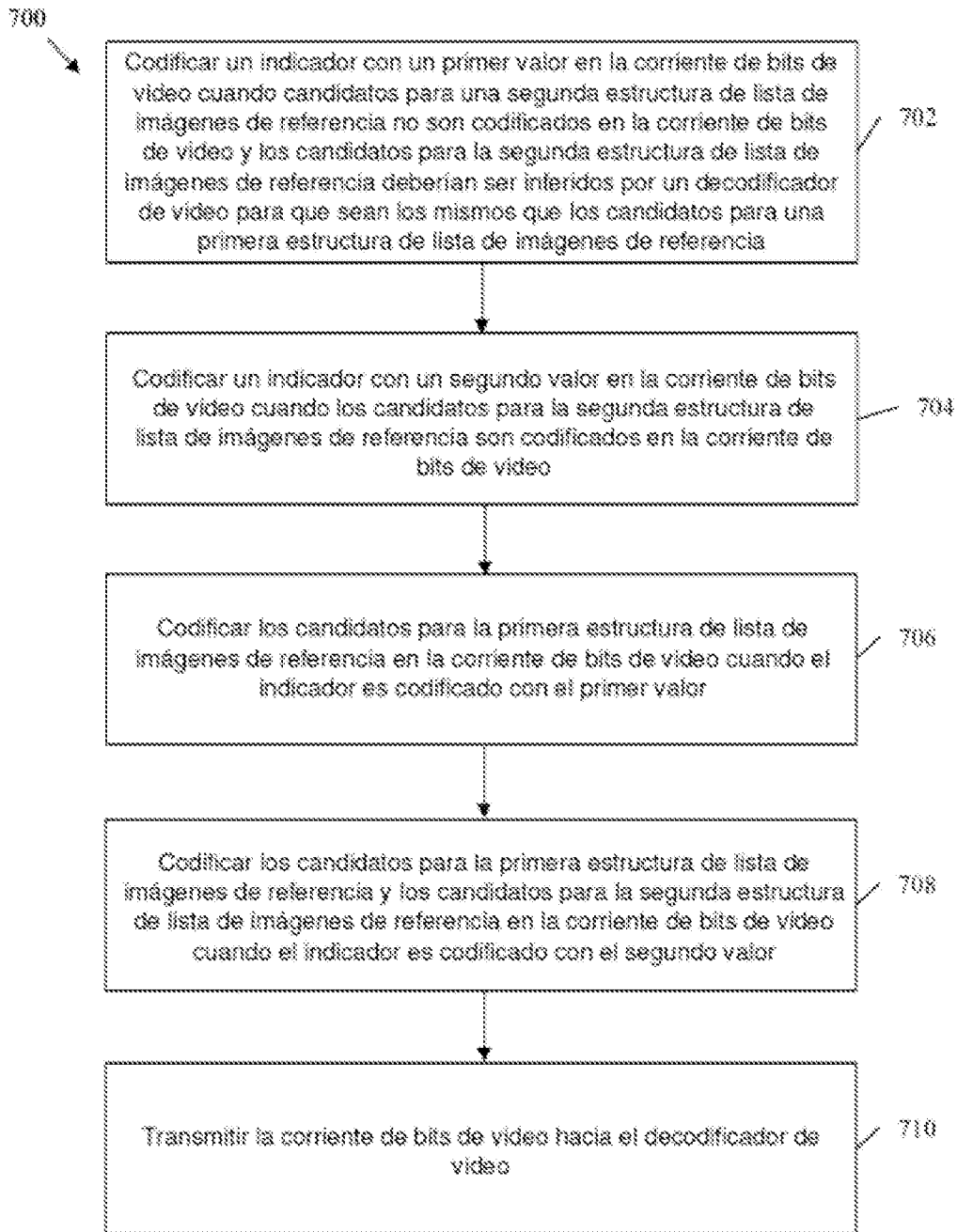


FIGURA 7

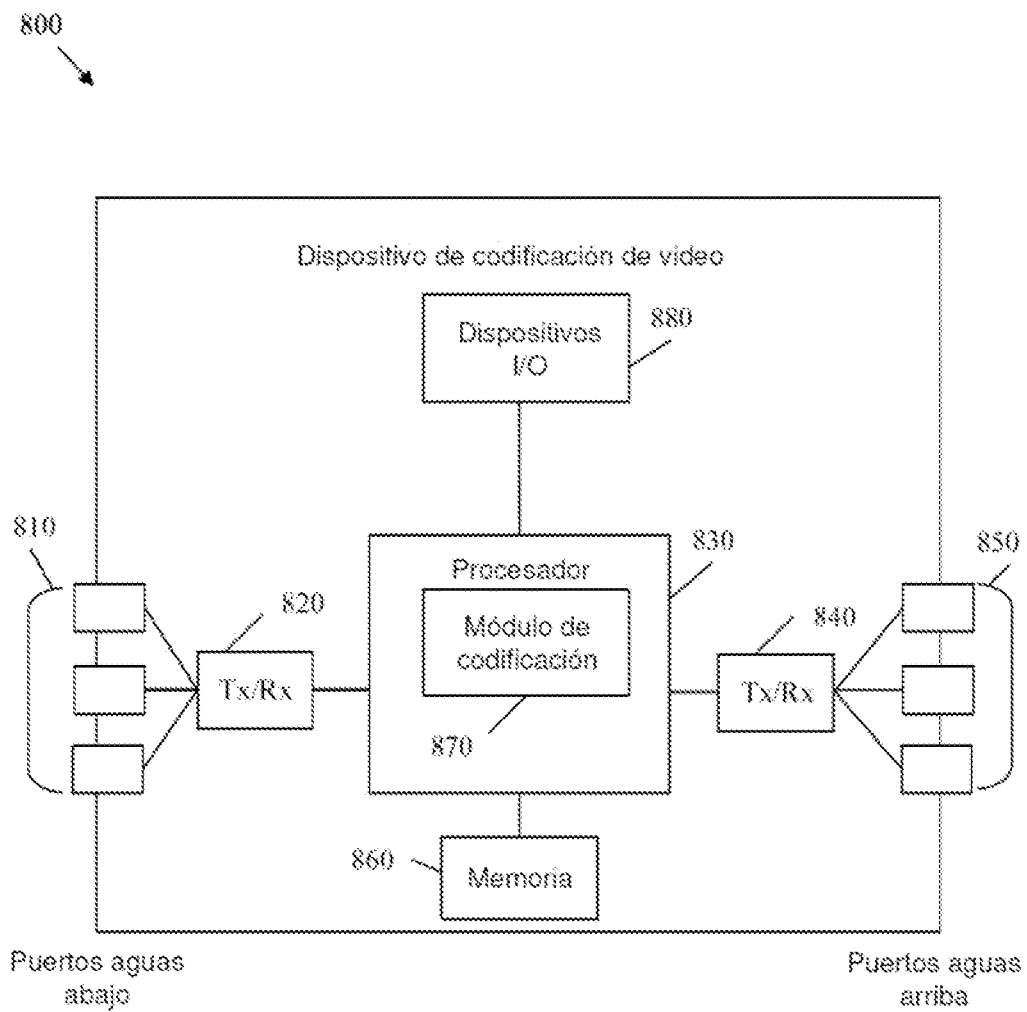


FIGURA 8

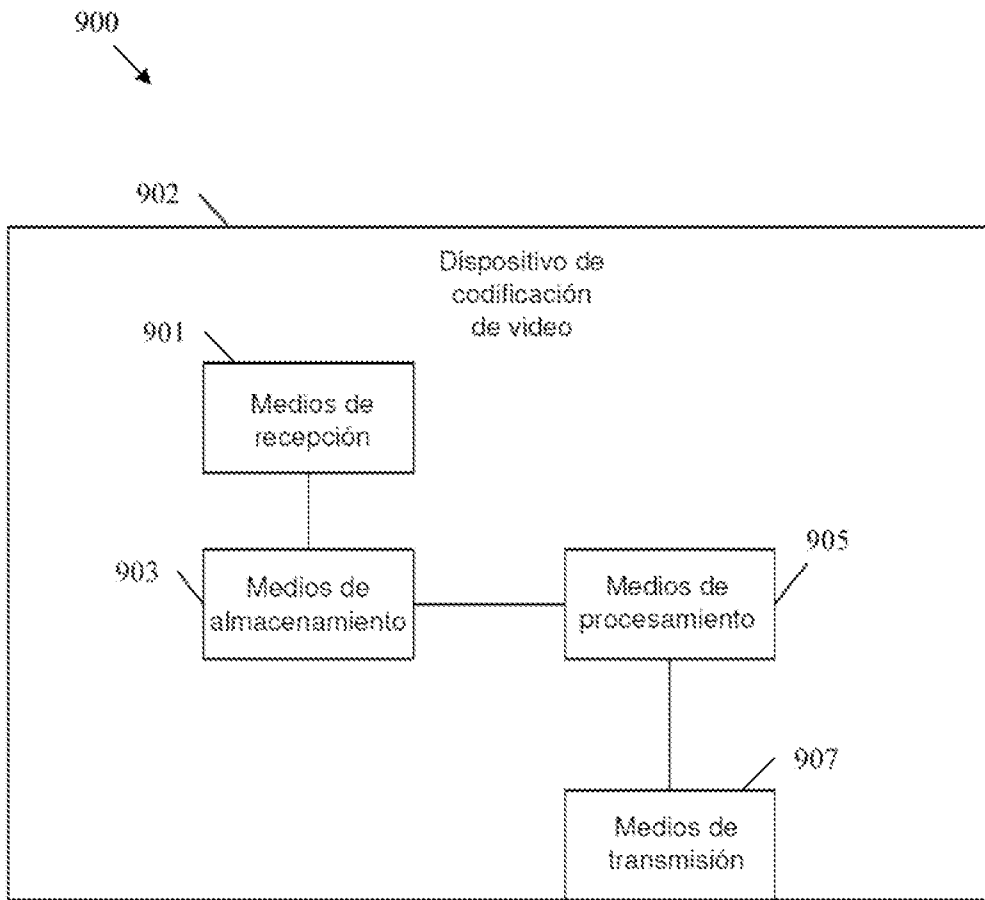


FIGURA 9