

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 935 979**

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/US2019/050857**

87 Fecha y número de publicación internacional: **19.03.2020 WO20056168**

96 Fecha de presentación y número de la solicitud europea: **12.09.2019 E 19858972 (3)**

97 Fecha y número de publicación de la concesión europea: **07.12.2022 EP 3847804**

54 Título: **Señalización de índices 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:

13.03.2023

73 Titular/es:

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

72 Inventor/es:

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

74 Agente/Representante:

FERNÁNDEZ POU, Felipe

ES 2 935 979 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

Señalización de índices para estructuras de listas de imágenes de referencia

5 Campo técnico

En general, la presente divulgación describe técnicas para mejorar la eficacia de la señalización en la gestión de imágenes de referencia en la codificación de vídeo. Más específicamente, esta divulgación describe técnicas para mejorar la señalización para la construcción de listas de imágenes de referencia y el marcado de imágenes de referencia que se basa directamente en las listas de imágenes de referencia.

Antecedentes

La cantidad de datos de vídeo necesaria para representar incluso un vídeo relativamente corto puede ser considerable, lo que puede provocar dificultades cuando los datos se van a transmitir o comunicar de otro modo a través de una red de comunicaciones con una capacidad de ancho de banda limitada. Por ello, los datos de vídeo suelen comprimirse antes de ser comunicados a través de las redes de telecomunicaciones modernas. El tamaño de un vídeo también puede ser un problema cuando se almacena en un dispositivo de almacenamiento, ya que los recursos de memoria pueden ser limitados. Los dispositivos de compresión de vídeo suelen utilizar software y/o hardware en la fuente para codificar los datos de vídeo antes de su transmisión o almacenamiento, disminuyendo así la cantidad de datos necesarios para representar imágenes de vídeo digital. A continuación, los datos comprimidos son recibidos en el destino por un dispositivo de descompresión de vídeo que decodifica los datos de vídeo. Con unos recursos de red limitados y una demanda cada vez mayor de una mayor calidad de vídeo, son deseables unas técnicas de compresión y descompresión mejoradas que mejoren la relación de compresión sin sacrificar apenas la calidad de la imagen. HANNUKSELA (NOKIA) M M ET AL: "AHG21: On reference picture list construction and reference picture marking", 7.JCT-VC MEETING; 20111121 -20111130; GENEVA; (JOINT COLLABORATIVE TEAM ON VIDEO CODING OF ISO/IEC JTC1/SC29/WG11 AND ITU-T SG.16), no. JCTVC-G643 21 de noviembre de 2011 (2011-11-21), XP030230380, divulga un proceso de decodificación para generar una lista de imágenes de referencia con base en estructuras de sintaxis de descripción de lista de imágenes de referencia.

La solicitud de patente PCT No. WO2012033327A2 divulga 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. No. US2013259130A1 divulga un proceso de decodificación donde el decodificador de vídeo puede recibir, desde el flujo de bits codificado, la bandera que indica que RefPicList0 y RefPicList1 son sustancialmente equivalentes, y puede esperar recibir únicamente los parámetros de predicción ponderados para una de las listas.

40 Resumen

La presente invención se describe en las reivindicaciones adjuntas. A continuación, las partes de la descripción y del dibujo que se refieren a realizaciones, ejemplos o implementaciones, que no están cubiertas por las reivindicaciones, no se presentan como realizaciones de la invención, sino como ejemplos ilustrativos útiles para comprender la invención.

Se proporciona un primer aspecto relacionado con un método de decodificación de un flujo de bits de vídeo codificado implementado por un decodificador de vídeo, de acuerdo con la reivindicación 1.

El método proporciona técnicas que simplifican y hacen más eficiente el proceso de codificación. Mediante el uso de una bandera para especificar si un índice de una segunda estructura de lista de imágenes de referencia puede inferirse que es el mismo que un índice de una primera estructura de lista de imágenes de referencia, el codificador / decodificador (también conocido como "códec") en la codificación de vídeo se mejora (por ejemplo, utiliza menos bits, exige menos ancho de banda, es más eficiente, etc.) en relación con los códecs actuales. En la práctica, el proceso de codificación de vídeo mejorado ofrece al usuario una mejor experiencia cuando envía, recibe y/o visualiza vídeos.

En una primera forma de implementación del método de acuerdo con el primer aspecto como tal, la bandera se designa rpl1_idx_present_flag.

60

En una segunda forma de implementación del método de acuerdo con el primer aspecto como tal o cualquier forma de implementación precedente del primer aspecto, la bandera se incluye en un conjunto de parámetros de imagen (PPS) del flujo de bits de vídeo codificado.

En una tercera forma de implementación del método de acuerdo con el primer aspecto como tal o cualquier

forma de implementación precedente del primer aspecto, la bandera se incluye en un conjunto de parámetros de secuencia (SPS) del flujo de bits de vídeo codificado.

5 En una cuarta forma de implementación del método de acuerdo con el primer aspecto como tal o cualquier forma de implementación precedente del primer aspecto, la primera estructura de lista de imágenes de referencia se incluye en la cabecera de fragmento del flujo de bits de vídeo codificado.

10 En una quinta forma de implementación del método de acuerdo con el primer aspecto como tal o cualquier forma de implementación precedente del primer aspecto, la bandera se incluye en un conjunto de parámetros de imagen (PPS) del flujo de bits de vídeo codificado y la primera estructura de lista de imágenes de referencia se incluye en la cabecera de fragmento del flujo de bits de vídeo codificado.

15 En una sexta forma de implementación del método de acuerdo con el primer aspecto como tal o cualquier forma de implementación precedente del primer aspecto, el primer valor de la bandera es uno (1).

En una séptima forma de implementación del método de acuerdo con el primer aspecto como tal o cualquier forma de implementación precedente del primer aspecto, una `ref_pic_list_sps_flag[1]` y una `ref_pic_list_idx[1]` no se incluyen en la cabecera de fragmento cuando el primer valor de la bandera es uno (1).

20 En una octava forma de implementación del método de acuerdo con el primer aspecto como tal o cualquier forma de implementación precedente del primer aspecto, el segundo valor de la bandera es cero (0).

25 En una novena forma de implementación del método de acuerdo con el primer aspecto como tal o cualquier forma de implementación precedente del primer aspecto, una `ref_pic_list_sps_flag[0]` y una `ref_pic_list_idx[0]` se incluyen en la cabecera de fragmento cuando el segundo valor de la bandera es cero (0).

Se proporciona un segundo aspecto relacionado un método de codificación de un flujo de bits de vídeo implementado por un codificador de vídeo, de acuerdo con la reivindicación 8.

30 El método proporciona técnicas que simplifican y hacen más eficiente el proceso de codificación. Mediante el uso de una bandera para especificar si un índice de una segunda estructura de lista de imágenes de referencia puede inferirse que es el mismo que un índice de una primera estructura de lista de imágenes de referencia, el codificador / decodificador (también conocido como "códec") en la codificación de vídeo se mejora (por ejemplo, utiliza menos bits, exige menos ancho de banda, es más eficiente, etc.) en relación con los códecs
35 actuales. En la práctica, el proceso de codificación de vídeo mejorado ofrece al usuario una mejor experiencia cuando envía, recibe y/o visualiza vídeos.

40 En una primera forma de implementación del método de acuerdo con el segundo aspecto como tal, la bandera se designa `rpl1_idx_present_flag`.

En una segunda forma de implementación del método de acuerdo con el segundo aspecto como tal o cualquier forma de implementación precedente del segundo aspecto, la bandera se codifica en un conjunto de parámetros de imagen (PPS) del flujo de bits de vídeo codificado.

45 En una tercera forma de implementación del método de acuerdo con el segundo aspecto como tal o cualquier forma de implementación precedente del segundo aspecto, la primera estructura de lista de imágenes de referencia se codifica en la cabecera de fragmento del flujo de bits de vídeo codificado.

50 En una cuarta forma de implementación del método de acuerdo con el segundo aspecto como tal o cualquier forma de implementación precedente del segundo aspecto, la primera estructura de lista de imágenes de referencia y la segunda lista de imágenes de referencia se codifican en la cabecera de fragmento del flujo de bits de vídeo codificado.

55 En una quinta forma de implementación del método de acuerdo con el segundo aspecto como tal o cualquier forma de implementación precedente del segundo aspecto, el primer valor de la bandera es uno (1) y el segundo valor de la bandera es cero (0).

60 En una sexta forma de implementación del método de acuerdo con el segundo aspecto como tal o cualquier forma de implementación precedente del segundo aspecto, una `ref_pic_list_sps_flag[1]` y una `ref_pic_list_idx[1]` no se incluyen en la cabecera de fragmento cuando el primer valor de la bandera es uno (1).

65 Un tercer aspecto se relaciona con un dispositivo de decodificación que comprende un receptor configurado para recibir un flujo de bits de vídeo codificado; una memoria acoplada al receptor, la memoria almacena instrucciones; y un procesador acoplado a la memoria, el procesador configurado para ejecutar las instrucciones almacenadas en la memoria para hacer que el procesador: analice, a partir del flujo de bits de

vídeo codificado, una bandera; analice, a partir del flujo de bits de vídeo codificado, una primera estructura de lista de imágenes de referencia; determine que un índice de una segunda estructura de lista de imágenes de referencia no está presente en una cabecera de fragmento del flujo de bits de vídeo codificado e infiera que el índice de la segunda estructura de lista de imágenes de referencia es el mismo que un índice de la primera estructura de lista de imágenes de referencia cuando la bandera tiene un primer valor; determinar que el índice de la segunda estructura de lista de imágenes de referencia está presente en la cabecera de fragmento cuando la bandera tiene un segundo valor; generar una lista de imágenes de referencia utilizando al menos una de las estructuras de lista de imágenes de referencia primera o segunda; y realizar una interpolación basada en la lista de imágenes de referencia para generar un bloque reconstruido.

El dispositivo de decodificación proporciona técnicas que simplifican y hacen más eficiente el proceso de codificación. Mediante el uso de una bandera para especificar si un índice de una segunda estructura de lista de imágenes de referencia puede inferirse que es el mismo que un índice de una primera estructura de lista de imágenes de referencia, se mejora el codificador/decodificador (también denominado "códec") en la codificación de vídeo (por ejemplo, utiliza menos bits, exige menos ancho de banda, es más eficiente, etc.) en relación con los códecs actuales. En la práctica, el proceso de codificación de vídeo mejorado ofrece al usuario una mejor experiencia cuando envía, recibe o visualiza vídeos.

En una primera forma de implementación del dispositivo de decodificación según el tercer aspecto como tal, el dispositivo de decodificación incluye una pantalla configurada para mostrar una imagen generada utilizando el bloque reconstruido.

En una segunda forma de implementación del dispositivo decodificador según el tercer aspecto como tal o cualquier forma de implementación precedente del tercer aspecto, la bandera se designa `rpl1_idx_present_flag`.

En una tercera forma de implementación del dispositivo decodificador según el tercer aspecto como tal o cualquier forma de implementación precedente del tercer aspecto, la bandera se incluye en un conjunto de parámetros de imagen (PPS) del flujo de bits de vídeo codificado.

En una cuarta forma de implementación del dispositivo de decodificación según el tercer aspecto como tal o cualquier forma de implementación precedente del tercer aspecto, la primera estructura de lista de imágenes de referencia se incluye en la cabecera de fragmento del flujo de bits de vídeo codificado.

En una quinta forma de implementación del dispositivo decodificador según el tercer aspecto como tal o cualquier forma de implementación precedente del tercer aspecto, el primer valor de la bandera es uno (1) y el segundo valor de la bandera es cero (0).

En una sexta forma de implementación del dispositivo de decodificación según el tercer aspecto como tal o cualquier forma de implementación precedente del tercer aspecto, una bandera `ref_pic_list_sps[1]` y una `ref_pic_list_idx[1]` no se incluyen en la cabecera de fragmento cuando el primer valor de la bandera es uno (1).

Un cuarto aspecto se relaciona con un dispositivo de codificación que comprende un procesador configurado para: codificar una bandera con un primer valor en el flujo de bits de vídeo cuando un índice de una segunda estructura de lista de imágenes de referencia no está codificado en una cabecera de fragmento del flujo de bits de vídeo y un decodificador de vídeo debería deducir que el índice de la segunda estructura de lista de imágenes de referencia es el mismo que el índice de una primera estructura de lista de imágenes de referencia; codificar una bandera con un segundo valor en el flujo de bits de vídeo cuando el índice de la segunda estructura de lista de imágenes de referencia está codificado en la cabecera de fragmento del flujo de bits de vídeo; codificar la primera estructura de lista de imágenes de referencia en el flujo de bits de vídeo cuando la bandera se codifica con el primer valor; codificar la primera estructura de lista de imágenes de referencia y la segunda estructura de lista de imágenes de referencia en el flujo de bits de vídeo cuando la bandera se codifica con el segundo valor; y un transmisor acoplado al procesador, el transmisor configurado para transmitir el flujo de bits de vídeo hacia el decodificador de vídeo.

El dispositivo de codificación proporciona técnicas que simplifican y hacen más eficiente el proceso de codificación. Mediante el uso de un indicador para especificar si un índice de una segunda estructura de lista de imágenes de referencia puede inferirse que es el mismo que un índice de una primera estructura de lista de imágenes de referencia, se mejora el codificador/decodificador (también denominado "códec") en la codificación de vídeo (por ejemplo, utiliza menos bits, exige menos ancho de banda, es más eficiente, etc.) en relación con los códecs actuales. En la práctica, el proceso de codificación de vídeo mejorado ofrece al usuario una mejor experiencia cuando envía, recibe o visualiza vídeos.

En una primera forma de implementación del dispositivo de codificación de acuerdo con el cuarto aspecto como

tal, la bandera se designa rpl1_idx_present_flag.

5 En una segunda forma de implementación del dispositivo de codificación de acuerdo con el cuarto aspecto como tal o cualquier forma de implementación precedente del cuarto aspecto, la bandera se codifica en un conjunto de parámetros de imagen (PPS) del flujo de bits de vídeo codificado.

10 En una tercera forma de implementación del dispositivo de codificación de acuerdo con el cuarto aspecto como tal o cualquier forma de implementación precedente del cuarto aspecto, la primera estructura de lista de imágenes de referencia se codifica en la cabecera de fragmento del flujo de bits de vídeo codificado.

15 En una cuarta forma de implementación del dispositivo de codificación de acuerdo con el cuarto aspecto como tal o cualquier forma de implementación precedente del cuarto aspecto, el primer valor de la bandera es uno (1) y el segundo valor de la bandera es cero (0).

20 En una quinta forma de implementación del dispositivo de codificación de acuerdo con el cuarto aspecto como tal o cualquier forma de implementación precedente del cuarto aspecto, una ref_pic_list_sps_flag[1] y una ref_pic_list_idx[1] no se incluyen en la cabecera de fragmento cuando el primer valor de la bandera es uno (1).

25 Un quinto aspecto se relaciona con un aparato de codificación que comprende: un receptor configurado para recibir un flujo de bits para decodificar; un transmisor acoplado al receptor, el transmisor configurado para transmitir una imagen decodificada a una pantalla; una memoria acoplada a por lo menos uno de los receptores o el transmisor, la memoria configurada para almacenar instrucciones; y un procesador acoplado a la memoria, el procesador configurado para ejecutar las instrucciones almacenadas en la memoria para realizar el método en cualquiera de las realizaciones divulgadas en el presente documento.

30 El aparato de codificación proporciona técnicas que simplifican y hacen más eficiente el proceso de codificación. Mediante el uso de un indicador para especificar si un índice de una segunda estructura de lista de imágenes de referencia puede inferirse que es el mismo que un índice de una primera estructura de lista de imágenes de referencia, el codificador / decodificador (también denominado "códec") en la codificación de vídeo se mejora (por ejemplo, utiliza menos bits, exige menos ancho de banda, es más eficiente, etc.) en relación con los códecs actuales. En la práctica, el proceso de codificación de vídeo mejorado ofrece al usuario una mejor experiencia cuando envía, recibe o visualiza vídeos.

35 Un séptimo aspecto se relaciona con un sistema que comprende un codificador que incluye un dispositivo de codificación en cualquiera de las realizaciones divulgadas en el presente documento; y un decodificador en comunicación con el codificador, incluyendo el decodificador un dispositivo de decodificación en cualquiera de las realizaciones divulgadas en el presente documento.

40 El sistema proporciona técnicas que simplifican y hacen más eficiente el proceso de codificación. Mediante el uso de un indicador para especificar si un índice de una segunda estructura de lista de imágenes de referencia puede inferirse que es el mismo que un índice de una primera estructura de lista de imágenes de referencia, se mejora el codificador/decodificador (también denominado "códec") en la codificación de vídeo (por ejemplo, utiliza menos bits, exige menos ancho de banda, es más eficiente, etc.) en relación con los códecs actuales.
45 En la práctica, el proceso de codificación de vídeo mejorado ofrece al usuario una mejor experiencia cuando envía, recibe o visualiza vídeos.

50 Un octavo aspecto se relaciona con medios para codificar que comprenden medios de recepción configurados para recibir un flujo de bits a decodificar; medios de transmisión acoplados a los medios de recepción, los medios de transmisión configurados para transmitir una imagen decodificada a un medio de visualización; medios de almacenamiento acoplados a al menos uno de los medios de recepción o los medios de transmisión, los medios de almacenamiento configurados para almacenar instrucciones; y medios de procesamiento acoplados a los medios de almacenamiento, los medios de procesamiento configurados para ejecutar las instrucciones almacenadas en los medios de almacenamiento para realizar el método en cualquiera de las realizaciones divulgadas en el presente documento.

60 Los medios de codificación proporcionan técnicas que simplifican y hacen más eficiente el proceso de codificación. Mediante el uso de un indicador para especificar si un índice de una segunda estructura de lista de imágenes de referencia puede inferirse que es el mismo que un índice de una primera estructura de lista de imágenes de referencia, el codificador/decodificador (también denominado "códec") en la codificación de vídeo mejora (por ejemplo, utiliza menos bits, exige menos ancho de banda, es más eficiente, etc.) en relación con los códecs actuales. En la práctica, el proceso de codificación de vídeo mejorado ofrece al usuario una mejor experiencia cuando envía, recibe o visualiza vídeos.

Breve descripción de los dibujos

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

5

La FIG. 1 es un diagrama de bloques que ilustra un ejemplo de sistema de codificación que puede utilizar técnicas de predicción bilateral.

10

La FIG. 2 es un diagrama de bloques que ilustra un ejemplo de codificador de vídeo que puede implementar técnicas de predicción bilateral.

La FIG. 3 es un diagrama de bloques que ilustra un ejemplo de decodificador de vídeo que puede implementar técnicas de predicción bilateral.

15

La FIG. 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.

La FIG. 5 es un diagrama esquemático de una realización de un flujo de bits de vídeo.

20

La FIG. 6 es una realización de un método de decodificación de un flujo de bits de vídeo codificado.

La FIG. 7 es una realización de un método de codificación de un flujo de bits de vídeo.

La FIG. 8 es un diagrama esquemático de un dispositivo de codificación de vídeo.

25

FIG. 9 es un diagrama esquemático de una realización de un medio de codificación.

Descripción detallada

30

A continuación se indican diversos acrónimos empleados en el presente documento: Memoria intermedia de imágenes decodificadas (DPB), actualización instantánea de decodificación (IDR), punto de acceso intraaleatorio (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 brutos (RBSP), conjunto de parámetros de secuencia (SPS) y borrador de trabajo (WD).

35

La FIG. 1 es un diagrama de bloques que ilustra un ejemplo de sistema 10 de codificación que puede utilizar técnicas de codificación de vídeo tal como se describen en el presente documento. Como se muestra en la FIG. 1, el sistema 10 de codificación incluye un dispositivo 12 de origen que proporciona datos de vídeo codificados para ser decodificados en un momento posterior por un dispositivo 14 de destino. En particular, el dispositivo 12 de origen puede proporcionar datos de vídeo al dispositivo 14 de destino a través de un medio 16 legible por ordenador. El dispositivo 12 de origen y el dispositivo 14 de destino pueden incluir una amplia gama de dispositivos, como ordenadores de sobremesa, ordenadores portátiles, tabletas, decodificadores, teléfonos inteligentes, televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivos de transmisión de vídeo o similares. En algunos casos, el dispositivo 12 de origen y el dispositivo 14 de destino pueden estar equipados para la comunicación inalámbrica.

40

El dispositivo 14 de destino puede recibir los datos de vídeo codificados a decodificar 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 vídeo codificados desde el dispositivo 12 de origen al dispositivo 14 de destino. En un ejemplo, el medio 16 legible por ordenador puede comprender un medio de comunicación que permita al dispositivo 12 de origen transmitir datos de vídeo codificados directamente al dispositivo 14 de destino en tiempo real. Los datos de vídeo codificados pueden ser modulados de acuerdo con un estándar de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitidos al dispositivo 14 de destino. El medio de comunicación puede comprender cualquier protocolo de comunicación inalámbrica. El medio de comunicación puede comprender cualquier medio de comunicación inalámbrico o por cable, tal como un espectro de radiofrecuencia (RF) o una o más líneas físicas de transmisión. El medio de comunicación puede formar parte de una red con base en paquetes, tal como una red de área local, una red de área extensa o una red global como Internet. El medio de comunicación puede incluir enrutadores, conmutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo 12 de origen hasta el dispositivo 14 de destino.

50

55

60

En algunos ejemplos, los datos codificados pueden enviarse desde la interfaz 22 de salida a un dispositivo de almacenamiento. Del mismo modo, se puede acceder a los datos codificados desde el dispositivo de almacenamiento mediante la interfaz de entrada. El dispositivo de almacenamiento puede incluir cualquiera de una variedad de medios de almacenamiento de datos distribuidos o de acceso local, tal como un disco duro,

65

discos Blu-ray, discos de vídeo digital (DVD), memorias de sólo lectura de discos compactos (CD-ROM), memoria flash, memoria volátil o no volátil, o cualquier otro medio de almacenamiento digital adecuado para almacenar datos de vídeo codificados. En otro ejemplo, el dispositivo de almacenamiento puede corresponder a un servidor de archivos u otro dispositivo de almacenamiento intermedio que pueda almacenar el vídeo codificado generado por el dispositivo 12 de origen. El dispositivo 14 de destino puede acceder a los datos de vídeo almacenados desde el dispositivo de almacenamiento mediante transmisión o descarga. El servidor de archivos puede ser cualquier tipo de servidor capaz de almacenar datos de vídeo codificados y transmitir dichos datos de vídeo codificados al dispositivo 14 de destino. Ejemplos de servidores de archivos incluyen un servidor web (por ejemplo, para un sitio web), un servidor de protocolo de transferencia de archivos (FTP), dispositivos de almacenamiento conectados a la red (NAS), o una unidad de disco local. El dispositivo 14 de destino puede acceder a los datos de vídeo codificados a través de cualquier conexión de datos estándar, incluida una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión Wi-Fi), una conexión por cable (por ejemplo, línea de abonado digital (DSL), módem de cable, etc.), o una combinación de ambos que sea adecuada para acceder a los datos de vídeo codificados almacenados en un servidor de archivos. La transmisión de datos de vídeo codificados desde el dispositivo de almacenamiento puede ser una transmisión de flujo continuo, una transmisión de descarga o una combinación de ambas.

Las técnicas de la presente divulgación no se limitan necesariamente a aplicaciones o entornos inalámbricos. Las técnicas pueden aplicarse a la codificación de vídeo en apoyo de cualquiera de una variedad de aplicaciones multimedia, tales como emisiones de televisión por aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de vídeo por Internet, tales como transmisión adaptativa dinámico sobre HTTP (DASH), vídeo digital que se codifica en un medio de almacenamiento de datos, decodificación de vídeo digital almacenado en un medio de almacenamiento de datos, u otras aplicaciones. En algunos ejemplos, el sistema 10 de codificación puede estar configurado para soportar la transmisión de vídeo unidireccional o bidireccional para soportar aplicaciones tales como transmisión de vídeo, reproducción de vídeo, difusión de vídeo y/o videotelefonía.

En el ejemplo de la FIG. 1, el dispositivo 12 de origen incluye la fuente 18 de vídeo, el codificador 20 de vídeo y la interfaz 22 de salida. El dispositivo 14 de destino incluye la interfaz 28 de entrada, el decodificador 30 de vídeo y el dispositivo 32 de visualización. De acuerdo con la presente divulgación, el codificador 20 de vídeo del dispositivo 12 de origen y/o el decodificador 30 de vídeo del dispositivo 14 de destino pueden configurarse para aplicar las técnicas de codificación de vídeo. En otros ejemplos, un dispositivo de origen y un dispositivo de destino pueden incluir otros componentes o disposiciones. Por ejemplo, el dispositivo 12 de origen puede recibir datos de vídeo de una fuente de vídeo externa, tal como una cámara externa. Del mismo modo, el dispositivo 14 de destino puede interactuar con un dispositivo de visualización externo, en lugar de incluir un dispositivo de visualización integrado.

El sistema 10 de codificación de la FIG. 1 es sólo un ejemplo. Las técnicas de codificación de vídeo pueden ser realizadas por cualquier dispositivo de codificación y/o decodificación de vídeo digital. Aunque las técnicas de esta divulgación generalmente son realizadas por un dispositivo de codificación de vídeo, las técnicas también pueden ser realizadas por un codificador/decodificador de vídeo, típicamente referido como un "CODEC". Además, las técnicas de la presente divulgación también pueden ser realizadas por un preprocesador de vídeo. El codificador y/o decodificador de vídeo puede ser una unidad de procesamiento gráfico (GPU) o un dispositivo similar.

El dispositivo 12 de origen y el dispositivo 14 de destino son meros ejemplos de tales dispositivos de codificación en los que el dispositivo 12 de origen genera datos de vídeo codificados para su transmisión al dispositivo 14 de destino. En algunos ejemplos, el dispositivo 12 de origen y el dispositivo 14 de destino pueden funcionar de forma sustancialmente simétrica, de manera que cada uno de los dispositivos 12, 14 de origen y destino incluye componentes de codificación y decodificación de vídeo. Por lo tanto, el sistema 10 de codificación puede soportar la transmisión de vídeo unidireccional o bidireccional entre los dispositivos de vídeo 12, 14, por ejemplo, para transmisión de vídeo, reproducción de vídeo, difusión de vídeo o videotelefonía.

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

En algunos casos, cuando la fuente 18 de vídeo es una cámara de vídeo, el dispositivo 12 de origen y el dispositivo 14 de destino pueden formar los denominados teléfonos con cámara o videoteléfonos. Sin embargo, como se ha mencionado anteriormente, las técnicas descritas en la presente divulgación pueden ser aplicables a la codificación de vídeo en general, y pueden aplicarse a aplicaciones inalámbricas y/o por cable. En cada caso, el vídeo capturado, precapturado o generado por ordenador puede ser codificado por el codificador 20 de vídeo. La información de vídeo codificada puede entonces ser emitida por la interfaz 22 de salida a un medio

16 legible por ordenador.

El medio 16 legible por ordenador puede incluir medios transitorios, tales como una transmisión de red inalámbrica o por cable, o medios de almacenamiento (es decir, medios de almacenamiento no transitorios), tal como un disco duro, una unidad flash, un disco compacto, un disco de vídeo digital, un disco Blu-ray u otros medios legibles por ordenador. En algunos ejemplos, un servidor de red (no mostrado) puede recibir datos de vídeo codificados del dispositivo 12 de origen y proporcionar los datos de vídeo codificados al dispositivo 14 de destino, por ejemplo, mediante transmisión de red. De forma similar, un dispositivo informático de una instalación de producción de medios, tal como una instalación de estampación de discos, puede recibir datos de vídeo codificados del dispositivo 12 de origen y producir un disco que contenga los datos de vídeo codificados. Por lo tanto, puede entenderse que el medio 16 legible por ordenador incluye uno o más medios legibles por ordenador de diversas formas, en diversos ejemplos.

La interfaz 28 de entrada del dispositivo 14 de destino recibe información del medio 16 legible por ordenador. La información del medio 16 legible por ordenador puede incluir información de sintaxis definida por el codificador 20 de vídeo, que también es utilizada por el decodificador 30 de vídeo, que incluye elementos de sintaxis que describen características y/o procesamiento de bloques y otras unidades codificadas, por ejemplo, grupos de imágenes (GOP). El dispositivo 32 de visualización muestra los datos de vídeo decodificados a un usuario, y puede comprender cualquiera de una variedad de dispositivos de visualización tales como un tubo de rayos catódicos (CRT), una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodo orgánico emisor de luz (OLED), u otro tipo de dispositivo de visualización.

El codificador 20 de vídeo y el decodificador 30 de vídeo pueden funcionar de acuerdo con una estándar de codificación de vídeo, tal como el estándar High Efficiency Video Coding (HEVC) actualmente en desarrollo, y pueden ajustarse al modelo de prueba HEVC (HM). Alternativamente, el codificador 20 de vídeo y el decodificador 30 de vídeo pueden operar de acuerdo con otros estándares propietarios o industriales, tales como el estándar H.264 del Sector de International Telecommunications Union Telecommunication Standardization Sector (UIT-T), alternativamente denominado Moving Picture Expert Group (MPEG)-4, Parte 10, Advance Video Coding (AVC), H.265/HEVC, o extensiones de dichos estándares. Sin embargo, las técnicas de la presente divulgación no se limitan a ninguna norma de codificación en particular. Otros ejemplos de estándares de codificación de vídeo incluyen MPEG-2 y ITU-T H.263. Aunque no se muestra en la FIG. 1, en algunos aspectos, el codificador 20 de vídeo y el decodificador 30 de vídeo pueden estar integrados cada uno con un codificador y decodificador de audio, y pueden incluir unidades multiplexoras-demultiplexoras (MUX-DEMUX) apropiadas, u otro hardware y software, para manejar la codificación tanto de audio como de vídeo en un flujo de datos común o en flujos de datos separados. Si procede, las unidades MUX-DEMUX pueden ajustarse al protocolo multiplexor H.223 de la UIT, o a otros protocolos tales como el protocolo de datagramas de usuario (UDP).

El codificador 20 de vídeo y el decodificador 30 de vídeo pueden implementarse como cualquiera de una variedad de circuitos de codificación adecuados, tales como uno o más microprocesadores, procesadores de señal digital (DSP), circuitos integrados de aplicación específica (ASIC), matrices de puertas programables en campo (FPGA), lógica discreta, software, hardware, firmware o cualquier combinación de los mismos. Cuando las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio adecuado no transitorio legible por ordenador y ejecutar las instrucciones en hardware utilizando uno o más procesadores para realizar las técnicas de la presente divulgación. Cada uno de los codificadores 20 de vídeo y decodificadores 30 de vídeo puede estar incluido en uno o más codificadores o decodificadores, cualquiera de los cuales puede estar integrado como parte de un codificador/decodificador combinado (CODEC) en un dispositivo respectivo. Un dispositivo que incluye el codificador 20 de vídeo y/o el decodificador 30 de vídeo puede comprender un circuito integrado, un microprocesador y/o un dispositivo de comunicación inalámbrica, tal como un teléfono móvil.

La FIG. 2 es un diagrama de bloques que ilustra un ejemplo de codificador 20 de vídeo que puede implementar técnicas de codificación de vídeo. El codificador 20 de vídeo puede realizar codificación intra e inter de bloques de vídeo dentro de rebanadas de vídeo. La intracodificación se basa en la predicción espacial para reducir o eliminar la redundancia espacial en el vídeo dentro de un cuadro o imagen de vídeo dado. La intercodificación se basa en la predicción temporal para reducir o eliminar la redundancia temporal en el vídeo dentro de cuadros o imágenes adyacentes de una secuencia de vídeo. El modo intra (modo I) puede referirse a cualquiera de los modos de codificación basados en el espacio. Los modos intermedios, como la predicción unidireccional (también conocida como predicción uni) (modo P) o la predicción bi (también conocida como predicción bi) (modo B), pueden referirse a cualquiera de los modos de codificación basados en el tiempo.

Como se muestra en la FIG. 2, el codificador 20 de vídeo recibe un bloque de vídeo actual dentro de un cuadro de vídeo a ser codificado. En el ejemplo de la FIG. 2, el codificador 20 de vídeo incluye la unidad 40 de selección de modo, memoria 64 de cuadro de referencia, totalizador 50, unidad 52 de procesamiento de transformación, unidad 54 de cuantificación, y unidad 56 de codificación de entropía. La unidad 40 de selección de modo, a su

- vez, incluye la unidad 44 de compensación de movimiento, la unidad 42 de estimación de movimiento, la unidad de intrapredicción (también conocida como intrapredicción) 46, y la unidad de partición 48. Para la reconstrucción de bloques de vídeo, el codificador 20 de vídeo también incluye la unidad 58 de cuantización inversa, la unidad 60 de transformación inversa y el totalizador 62. Un filtro de desbloqueo (no mostrado en la FIG. 2) también puede ser incluido para filtrar los límites de bloque para retirar artefactos de bloqueo del vídeo reconstruido. Si se desea, el filtro de desbloqueo filtraría típicamente la salida del totalizador 62. También se pueden usar filtros adicionales (en bucle o post bucle) además del filtro de desbloqueo. Tales filtros no se muestran por brevedad, pero si se desea, pueden filtrar la salida del totalizador 50 (como un filtro en bucle).
- 5
- 10 Durante el proceso de codificación, el codificador 20 de vídeo recibe un cuadro o fragmento de vídeo que debe codificarse. El cuadro o fragmento puede estar dividido en múltiples bloques de vídeo. La unidad 42 de estimación de movimiento y la unidad 44 de compensación de movimiento realizan la codificación interpredictiva del bloque de vídeo recibido en relación con uno o más bloques en uno o más cuadros de referencia para proporcionar predicción temporal. La unidad 46 de intrapredicción puede, alternativamente, realizar una codificación intrapredictiva del bloque de vídeo recibido en relación con uno o más bloques vecinos en el mismo cuadro o fragmento que el bloque a codificar para proporcionar predicción espacial. El codificador 20 de vídeo puede realizar múltiples pasadas de codificación, por ejemplo, para seleccionar un modo de codificación apropiado para cada bloque de datos de vídeo.
- 15
- 20 Además, la unidad 48 de partición puede particionar bloques de datos de vídeo en subbloques, con base en la evaluación de esquemas de partición previos en pases de codificación anteriores. Por ejemplo, la unidad 48 de partición puede inicialmente particionar un cuadro o fragmento en unidades de codificación más grandes (LCUs), y particionar cada una de las LCUs en subunidades de codificación (sub-CUs) basándose en el análisis de distorsión de velocidad (por ejemplo, optimización de distorsión de velocidad). La unidad 40 de selección de modo puede producir además una estructura de datos de árbol cuádruple indicativa de la partición de una LCU en sub-CUs. Las CU de nodo de hoja del árbol cuádruple pueden incluir una o más unidades de predicción (PU) y una o más unidades de transformación (TU).
- 25

30 La presente divulgación utiliza el término "bloque" para referirse a cualquiera de las CU, PU o TU, en el contexto de HEVC, o estructuras de datos similares en el contexto de otros estándares (por ejemplo, macrobloques y subbloques de los mismos en H.264/AVC). Una CU incluye un nodo de codificación, PUs y TUs asociadas al nodo de codificación. El tamaño de la CU corresponde al tamaño del nodo de codificación y tiene forma cuadrada. El tamaño de la CU puede oscilar entre 8×8 píxeles y el tamaño del bloque de árbol, con un máximo de 64×64 píxeles o superior. Cada CU puede contener una o más PUs y una o más TUs. Los datos de sintaxis asociados a una CU pueden describir, por ejemplo, la partición de la CU en una o más PUs. Los modos de partición pueden diferir entre sí si la CU está codificada en modo de salto o directo, en modo intrapredicción o en modo interpredicción (también conocido como interpredicción). Las PUs pueden ser particionadas para tener una forma no cuadrada. Los datos de sintaxis asociados a una CU también pueden describir, por ejemplo, la partición de la CU en una o más TU según un árbol cuádruple. Una UT puede tener forma cuadrada o no cuadrada (por ejemplo, rectangular).

35

40

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 inter codificado resultante al totalizador 50 para generar datos de bloque residuales y al totalizador 62 para reconstruir el bloque codificado para su uso como marco de referencia. La unidad 40 de selección de modo también proporciona elementos de sintaxis, tales como vectores de movimiento, indicadores intramodo, información de partición, y otra información de sintaxis de este tipo, a la unidad 56 de codificación de entropía.

45

La unidad 42 de estimación de movimiento y la unidad 44 de compensación de movimiento pueden estar altamente integradas, pero se ilustran por separado a efectos conceptuales. La estimación de movimiento, realizada por la unidad 42 de estimación de movimiento, es el proceso de generación de vectores de movimiento, que estiman el movimiento de los bloques de vídeo. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de una PU de un bloque de vídeo dentro de un cuadro o imagen de vídeo actual relativo a un bloque predictivo dentro de un cuadro de referencia (u otra unidad codificada) relativo al bloque actual que está siendo codificado dentro del cuadro actual (u otra unidad codificada). Un bloque predictivo es un bloque que se encuentra que coincide estrechamente con el bloque a codificar, en términos de diferencia de píxeles, que puede determinarse por suma de diferencia absoluta (SAD), suma de diferencia cuadrada (SSD), u otras métricas de diferencia. En algunos ejemplos, el codificador 20 de vídeo puede calcular valores para posiciones de píxeles subenteros de imágenes de referencia almacenadas en la memoria 64 de cuadros de referencia. Por ejemplo, el codificador 20 de vídeo puede interpolar valores de posiciones de un cuarto de píxel, posiciones de un octavo de píxel u otras posiciones fraccionarias de píxel de la imagen de referencia. Por lo tanto, la unidad 42 de estimación de movimiento puede realizar una búsqueda de movimiento relativa a las posiciones de píxel completo y posiciones de píxel fraccional y emitir un vector de movimiento con precisión de píxel fraccional.

50

55

60

65

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

La compensación de movimiento, realizada por la unidad 44 de compensación de movimiento, puede implicar la obtención 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 integradas funcionalmente en algunos ejemplos. Al recibir el vector de movimiento para el PU del bloque de vídeo actual, la unidad 44 de compensación de movimiento puede localizar el bloque predictivo al que apunta el vector de movimiento en una de las listas de imágenes de referencia. El totalizador 50 forma un bloque de vídeo residual restando los valores de píxel del bloque predictivo de los valores de píxel del bloque de vídeo actual que se está codificando, formando valores de diferencia de píxel, como se discute más adelante. En general, la unidad 42 de estimación de movimiento realiza la estimación de movimiento relativa a los componentes de luma, y la unidad 44 de compensación de movimiento utiliza vectores de movimiento calculados basándose en los componentes de luma tanto para los componentes de croma como para los componentes de luma. La unidad 40 de selección de modo también puede generar elementos de sintaxis asociados con los bloques de vídeo y la porción de vídeo para su uso por el decodificador 30 de vídeo en la decodificación de los bloques de vídeo de la porción de vídeo.

La unidad 46 de intrapredicción puede intrapredicir un bloque actual, como alternativa a la interpredicción realizada por la unidad 42 de estimación de movimiento y la unidad 44 de compensación de movimiento, como se ha descrito anteriormente. En particular, la unidad 46 de intrapredicción puede determinar un modo de intrapredicción a utilizar para codificar un bloque actual. En algunos ejemplos, la unidad 46 de intrapredicción puede codificar un bloque actual usando diversos modos de intrapredicción, por ejemplo, durante pases de codificación separados, y la unidad 46 de intrapredicción (o la unidad 40 de selección de modo, en algunos ejemplos) puede seleccionar un modo de intrapredicción apropiado a usar de los modos probados.

Por ejemplo, la unidad 46 de intrapredicción puede calcular los valores de distorsión de tasa utilizando un análisis de distorsión de tasa para los diversos modos de intrapredicción probados, y seleccionar el modo de intra-predicción que tenga las mejores características de distorsión de velocidad entre los modos probados. El análisis de distorsión de velocidad generalmente determina una cantidad de distorsión (o error) entre un bloque codificado y un bloque original no codificado que se codificó para producir el bloque codificado, así como una tasa de bits (es decir, un número de bits) utilizada para producir el bloque codificado. La unidad 46 de intrapredicción puede calcular ratios a partir de las distorsiones y tasas de los distintos bloques codificados para determinar qué modo de intrapredicción exhibe el mejor valor tasa-distorsión para el bloque.

Además, la unidad 46 de intrapredicción puede estar 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 intrapredicción y los otros modos DMM, por ejemplo, utilizando la optimización de la tasa de distorsión (RDO). Los datos de una imagen de textura correspondiente a un mapa de profundidad pueden almacenarse en la memoria 64 de cuadro de referencia. La unidad 42 de estimación de movimiento y la unidad 44 de compensación de movimiento también pueden configurarse para interpredicción de bloques de profundidad de un mapa de profundidad.

Después de seleccionar un modo de intrapredicción para un bloque (por ejemplo, un modo de intrapredicción convencional o uno de los modos DMM), la unidad 46 de intrapredicción puede proporcionar información indicativa del modo de intrapredicción seleccionado para el bloque a la unidad 56 de codificación de entropía. La unidad 56 de codificación de entropía puede codificar la información indicando el modo de intrapredicción seleccionado. El codificador 20 de vídeo puede incluir en los datos de configuración del flujo de bits transmitido, que pueden incluir una pluralidad de tablas de índice de modo de intrapredicción y una pluralidad de tablas de índice de modo de intra-predicción modificadas (también denominadas tablas de asignación de palabras de código), definiciones de contextos de codificación para diversos bloques e indicaciones de un modo de intrapredicción más probable, una tabla de índice de modo de intrapredicción y una tabla de índice de modo de intrapredicción modificada a utilizar para cada uno de los contextos.

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

La unidad 52 de procesamiento de transformación aplica una transformada, como una transformada coseno

discreta (DCT) o una transformada conceptualmente similar, al bloque residual, produciendo un bloque de vídeo que comprende valores de coeficiente de transformada residual. La unidad 52 de procesamiento de transformación puede realizar otras transformaciones conceptualmente similares a la DCT. También podrían utilizarse transformadas de onícula, transformadas de enteros, transformadas de subbandas u otros tipos de transformadas.

La unidad 52 de procesamiento de transformación aplica la transformación al bloque residual, produciendo un bloque de coeficientes de transformación residual. La transformación puede convertir la información residual de un dominio de valor de píxel a un dominio de transformación, tal como un dominio de frecuencia. La unidad 52 de procesamiento de transformación puede enviar los coeficientes de transformación resultantes a la unidad 54 de cuantificación. La unidad 54 de cuantificación cuantiza los coeficientes de transformación para reducir aún más la tasa de bits. El proceso de cuantificación puede reducir la profundidad de bits asociada con algunos o todos los coeficientes. El grado de cuantización puede modificarse ajustando un parámetro de cuantización. En algunos ejemplos, la unidad 54 de cuantificación puede entonces realizar un escaneo de la matriz incluyendo los coeficientes de transformación cuantificados. Alternativamente, la unidad 56 de codificación de entropía puede realizar la exploración.

Después de la cuantificación, la unidad 56 de codificación de entropía codifica los coeficientes de transformación cuantificados. Por ejemplo, la unidad 56 de codificación de entropía puede realizar codificación de longitud variable adaptable al contexto (CAVLC), codificación aritmética binaria adaptable al contexto (CABAC), codificación aritmética binaria adaptable al contexto basada en sintaxis (SBAC), codificación de entropía de partición de intervalo de probabilidad (PIPE) u otra técnica de codificación de entropía. En el caso de la codificación de entropía con base en el contexto, el contexto puede basarse en bloques vecinos. Después de la codificación de entropía por la unidad 56 de codificación de entropía, el flujo de bits codificado puede ser transmitido a otro dispositivo (por ejemplo, el decodificador 30 de vídeo) o archivado para su posterior transmisión o recuperación.

La unidad 58 de cuantización inversa y la unidad 60 de transformación inversa aplican la cuantización inversa y la transformación inversa, respectivamente, para reconstruir el bloque residual en el dominio del píxel, por ejemplo, para su uso posterior como 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 cuadros de la memoria 64 de cuadros 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 píxeles subenteros para su uso en la estimación de movimiento. El totalizador 62 añade 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 vídeo reconstruido para su almacenamiento en la memoria 64 de cuadros de referencia. El bloque de vídeo reconstruido puede utilizarse para la estimación de movimiento. El bloque de vídeo reconstruido puede ser utilizado por la unidad 42 de estimación de movimiento y la unidad 44 de compensación de movimiento como bloque de referencia para intercodificar un bloque en un cuadro de vídeo posterior.

La FIG. 3 es un diagrama de bloques que ilustra un ejemplo de decodificador 30 de vídeo que puede implementar técnicas de codificación de vídeo. En el ejemplo de la FIG. 3, el decodificador 30 de vídeo incluye una unidad 70 de decodificación de entropía, una unidad 72 de compensación de movimiento, una unidad 74 de intrapredicción, una unidad 76 de cuantización inversa, una unidad 78 de transformación inversa, una memoria 82 de cuadro de referencia y un totalizador 80. El decodificador 30 de vídeo puede, en algunos ejemplos, realizar un pase de decodificación generalmente recíproco al pase de codificación descrito con respecto al codificador 20 de vídeo (FIG. 2). La unidad 72 de compensación de movimiento puede generar datos de predicción con base en vectores de movimiento recibidos desde la unidad 70 de decodificación de entropía, mientras que la unidad 74 de intrapredicción puede generar datos de predicción con base en indicadores de modo de intrapredicción recibidos desde la unidad 70 de decodificación de entropía.

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

Cuando el fragmento de vídeo está codificado como un fragmento intracodificada (I), la unidad 74 de intrapredicción puede generar datos de predicción para un bloque de vídeo del fragmento de vídeo actual basándose en un modo de intrapredicción señalado y en datos de bloques previamente decodificados del cuadro o imagen actual. Cuando el cuadro de vídeo se codifica como un fragmento intercodificado (por ejemplo,

B, P, o GPB), la unidad 72 de compensación de movimiento produce bloques predictivos para un bloque de vídeo del fragmento de vídeo actual con base en los vectores de movimiento y otros elementos de sintaxis recibidos de la unidad 70 de decodificación de entropía. Los bloques predictivos pueden producirse a partir de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. El decodificador 30 de vídeo puede construir las listas de imágenes de referencia, Lista 0 y Lista 1, utilizando técnicas de construcción predeterminadas con base en imágenes de referencia almacenadas en la memoria 82 de cuadro de referencia.

La unidad 72 de compensación de movimiento determina la información de predicción para un bloque de vídeo del fragmento de vídeo actual analizando los vectores de movimiento y otros elementos de sintaxis, y utiliza la información de predicción para producir los bloques de predicción para el bloque de vídeo 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 interpredicción) utilizado para codificar los bloques de vídeo del fragmento de vídeo, un tipo de fragmento de interpredicción (por ejemplo, fragmento B, fragmento P, o fragmento GPB), información de construcción para una o más de las listas de imágenes de referencia para el fragmento, vectores de movimiento para cada bloque de vídeo intercodificado del fragmento, estado de interpredicción para cada bloque de vídeo intercodificado del fragmento, y otra información para decodificar los bloques de vídeo en el fragmento de vídeo actual.

La unidad 72 de compensación de movimiento también puede realizar interpolación con base en filtros de interpolación. La unidad 72 de compensación de movimiento puede usar filtros de interpolación como los usados por el codificador 20 de vídeo durante la codificación de los bloques de vídeo para calcular valores interpolados para subpíxeles enteros 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 vídeo a partir de los elementos de sintaxis recibidos y utilizar los filtros de interpolación para producir bloques predictivos.

Los datos de una imagen de textura correspondiente a un mapa de profundidad pueden almacenarse en la memoria 82 de cuadro de referencia. La unidad 72 de compensación de movimiento también puede estar configurada para interpredecir bloques de profundidad de un mapa de profundidad.

La compresión de imagen y vídeo ha experimentado un rápido crecimiento, lo que ha dado lugar a diversos estándares de codificación. Dichos estándares de codificación de vídeo incluyen International Telecommunication Union (ITU Telecommunications Standardization Sector (ITU-T) H.261, ISO/IEC Motion Picture Experts Group (MPEG)-1 Parte 2, UIT-T H.262 o ISO/IEC MPEG-2 Parte 2, ITU-T H.263, ISO/IEC MPEG-4 Parte 2, Advanced Video Coding (AVC), también conocida como ITU-T H.264 o ISO/IEC MPEG-4 Parte 10, y High Efficiency Video Coding (HEVC), también conocida como ITU-T H.265 o MPEG-H Parte 2. AVC incluye extensiones como Scalable Video Coding (SVC), Multiview Video Coding (MVC) y Multiview Video Coding plus Depth (MVC+D), y 3D AVC (3D-AVC). HEVC incluye extensiones como Scalable HEVC (SHVC), Multiview HEVC (MV-HEVC) y 3D HEVC (3D-HEVC).

Versatile Video Coding (VVC) es un nuevo estándar de codificación de vídeo que está desarrollando el equipo conjunto de expertos en vídeo (JVET) del UIT-T y la ISO/CEI. En el momento de redactar este documento, el último borrador de trabajo (WD) de VVC incluido en JVET-K1001-v1. El documento JVET JVET-K0325-v3 incluye una actualización de la sintaxis de alto nivel de VVC.

La presente divulgación describe técnicas destinadas a abordar el subdesarrollo del estándar WC. Sin embargo, las técnicas también pueden aplicarse a otras especificaciones de códecs de vídeo/medios.

Las técnicas de compresión de vídeo realizan predicciones espaciales (intraimagen) y/o temporales (interimagen) para reducir o eliminar la redundancia inherente a las secuencias de vídeo. Para la codificación de vídeo con base en bloques, un fragmento de vídeo (por ejemplo, una imagen de vídeo o una porción de una imagen de vídeo) puede dividirse en bloques de vídeo, que también pueden denominarse bloques de árbol, bloques de árbol de codificación (CTB), unidades de árbol de codificación (CTU), unidades de codificación (CU) y/o nodos de codificación. Los bloques de vídeo en un fragmento intracodificado (I) de una imagen son codificados usando predicción espacial con respecto a muestras de referencia en bloques vecinos en la misma imagen. Los bloques de vídeo en un fragmento intercodificado (P o B) de una imagen pueden usar predicción espacial con respecto a muestras de referencia en bloques vecinos en la misma imagen o predicción temporal con respecto a muestras de referencia en otras imágenes de referencia. Las imágenes pueden denominarse cuadros, y las imágenes de referencia, cuadros de referencia.

La predicción espacial o temporal da como resultado un bloque predictivo para un bloque a codificar. Los datos residuales representan las diferencias de píxeles entre el bloque original a codificar y el bloque predictivo. Un bloque intercodificado se codifica de acuerdo con un vector de movimiento que apunta a un bloque de muestras

de referencia que forma el bloque predictivo, y los datos residuales indican la diferencia entre el bloque codificado y el bloque predictivo. Un bloque intracodificado se codifica de acuerdo con un modo de intracodificación y los datos residuales. Para una mayor compresión, los datos residuales pueden transformarse del dominio del píxel a un dominio de transformación, dando lugar a coeficientes de transformación residuales, que luego pueden cuantificarse. Los coeficientes de transformación cuantificados, dispuestos inicialmente en una matriz bidimensional, pueden escanearse para producir un vector unidimensional de coeficientes de transformación, y puede aplicarse la codificación de entropía para lograr una compresión aún mayor.

En la especificación de un códec de vídeo, las imágenes se identifican con múltiples propósitos, incluyendo su uso como imagen de referencia en la interpredicción, para la salida de imágenes de la memoria intermedia de imágenes decodificadas (DPB), para el escalado de vectores de movimiento, para la predicción ponderada, etc. En AVC y HEVC, las imágenes pueden identificarse mediante el recuento de orden de imágenes (POC). En AVC y HEVC, las imágenes de la DPB pueden marcarse como "utilizadas para referencia a corto plazo", "utilizadas para referencia a largo plazo" o "no utilizadas para referencia". Una vez que una imagen se ha marcado como "no utilizada para referencia", ya no se puede utilizar para la predicción, y cuando ya no se necesita para la salida, se puede eliminar de la DPB.

En el AVC, hay dos tipos de imágenes de referencia, a corto y a largo plazo. Una imagen de referencia puede marcarse como "no utilizada para referencia" cuando deja de ser necesaria para la referencia de predicción. La conversión entre estos tres estados (a corto plazo, a largo plazo y no utilizada para referencia) se controla mediante el proceso de marcado de imágenes de referencia decodificadas. Existen dos mecanismos alternativos de marcado de imágenes de referencia decodificadas: el proceso de ventana deslizante implícita y el proceso de operación de control de gestión de memoria explícita (MMCO). El proceso de ventana deslizante marca una imagen de referencia a corto plazo como "no utilizada para referencia" cuando el número de cuadros de referencia es igual a un número máximo determinado (max_num_ref_frames en el SPS). Las imágenes de referencia a corto plazo se almacenan según el orden de entrada, de modo que las imágenes a corto plazo decodificadas más recientemente se conservan en la DPB.

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

En AVC, las operaciones de marcado de la imagen de referencia, así como los procesos de salida y eliminación de imágenes de la DPB, se realizan después de decodificar una imagen.

HEVC introduce un enfoque diferente para la gestión de imágenes de referencia, denominado conjunto de imágenes de referencia (RPS). La diferencia fundamental con el concepto RPS en comparación con MMCO/ventana deslizante de AVC es que para cada corte concreto se proporciona un conjunto completo de las imágenes de referencia que utiliza la imagen actual o cualquier imagen posterior. De este modo, se indica un conjunto completo de todas las imágenes que deben conservarse en la DPB para ser utilizadas por la imagen actual o futura. Esto difiere del esquema AVC, en el que sólo se señalan los cambios relativos en la DPB. Con el concepto RPS, no se necesita información de imágenes anteriores en el orden de decodificación para mantener el estado correcto de las imágenes de referencia en la DPB.

El orden de las operaciones de decodificación de imágenes y DPB en HEVC se modifica en comparación con AVC para aprovechar las ventajas de RPS y mejorar la resistencia a errores. En las operaciones de marcado y almacenamiento en búfer de imágenes AVC, tanto la salida como la eliminación de imágenes decodificadas de la DPB se aplican generalmente después de que se haya decodificado una imagen actual. En HEVC, se decodifica primero a partir de una cabecera de fragmento de la imagen actual y, a continuación, las operaciones de marcado de imágenes y de almacenamiento en búfer se aplican generalmente antes de decodificar la imagen actual.

Cada cabecera de fragmento en HEVC debe incluir parámetros para la señalización del RPS para la imagen que contiene los fragmentos. La única excepción es que no se señala ningún RPS para los fragmentos IDR, en su lugar se infiere que el RPS está vacío. Para los fragmentos I que no pertenecen a una imagen IDR, se puede proporcionar una RPS, aunque pertenezcan a una imagen I, ya que puede haber imágenes que sigan a la imagen I en el orden de decodificación que utilicen la interpredicción de imágenes que preceden a la imagen I en el orden de decodificación. El número de imágenes de una RPS no superará el límite de tamaño de DPB especificado en el elemento de sintaxis sps_max_dec_pic_buffering de la SPS.

Cada imagen está asociada a un valor POC que representa el orden de salida. Las cabeceras de fragmento contienen una palabra clave de longitud fija, pic_order_cnt_lsb, que representa los bits menos significativos del

valor POC completo, también conocido como POC LSB. La longitud de la palabra de código se indica en el SPS y puede estar comprendida entre 4 y 16 bits. El concepto RPS utiliza el POC para identificar las imágenes de referencia. Además de su propio valor POC, cada cabecera de fragmento contiene directamente o hereda del SPS una representación codificada de los valores POC (o los LSB) de cada imagen del RPS.

5

El RPS de cada imagen incluye cinco listas diferentes de imágenes de referencia, también denominadas 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 de salida, y que pueden utilizarse en la interpredicción 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 suceden a la imagen actual en el orden de salida y que pueden utilizarse en la interpredicción de la imagen actual. RefPicSetStFoll incluye todas las imágenes de referencia a corto plazo que pueden utilizarse en la interpredicción de una o varias de las imágenes que siguen a la imagen actual en el orden de decodificación y que no se utilizan en la interpredicción de la imagen actual. RefPicSetLtCurr incluye todas las imágenes de referencia a largo plazo que pueden utilizarse en la interpredicción de la imagen actual. RefPicSetLtFoll incluye todas las imágenes de referencia a largo plazo que pueden utilizarse en la interpredicción de una o varias de las imágenes que siguen a la imagen actual en el orden de decodificación y que no se utilizan en la interpredicción de la imagen actual.

10

15

20

25

El RPS se señala utilizando hasta tres bucles que iteran sobre diferentes tipos de imágenes de referencia: imágenes de referencia a corto plazo con un valor de POC inferior al de la imagen actual, imágenes de referencia a corto plazo con un valor de POC superior al de la imagen actual e imágenes de referencia a largo plazo. Además, se envía un indicador (`used_by_curr_pic_X_flag`) para cada imagen de referencia que indica si la imagen de referencia es utilizada como 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).

30

35

La FIG. 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 FIG. 4, la imagen actual B14 contiene exactamente una imagen en cada uno de los cinco subconjuntos 402 del RPS. P8 es la imagen en el subconjunto 402 referido como RefPicSetStCurrBefore porque la imagen está antes en el orden de salida y es usada por B14. P12 es la imagen del subconjunto 402 denominada RefPicSetStCurrAfter porque la imagen está después en el orden de salida y es utilizada por B14. P13 es la imagen en el subconjunto 402 referida como RefPicSetStFoll porque la imagen es una imagen de referencia a corto plazo no utilizada por B14 (pero debe mantenerse en la DPB ya que es utilizada por B15). P4 es la imagen del subconjunto 402 denominada RefPicSetLtCurr porque se trata de una imagen de referencia a largo plazo utilizada por B14. I0 es la imagen del subconjunto 402 denominada RefPicSetLtFoll, ya que se trata de una imagen de referencia a largo plazo no utilizada por la imagen actual (pero que debe mantenerse en la DPB, ya que es utilizada por B15).

40

45

La parte a corto plazo de la RPS 400 puede incluirse directamente en la cabecera de fragmento. Alternativamente, la cabecera puede contener sólo un elemento sintáctico que represente un índice, haciendo referencia a una lista predefinida de RPSs enviados en el SPS activo. La parte a corto plazo de la RPS 400 puede señalizarse utilizando uno de dos esquemas diferentes; Inter RPS, como se describe a continuación, o Intra RPS, como se describe aquí. Cuando se usa Intra RPS, `num_negative_pics` y `num_positive_pics` se señalizan representando la longitud de dos listas diferentes de imágenes de referencia. Estas listas contienen las imágenes de referencia con diferencia de POC negativa y diferencia de POC positiva en comparación con la imagen actual, respectivamente. Cada elemento de estas listas se codifica con un código de longitud variable que representa la diferencia de valor POC en relación con el elemento anterior de la lista menos uno. Para la primera imagen de cada lista, la señalización es relativa al valor POC de la imagen actual menos uno.

50

55

60

Al codificar los RPS recurrentes en el conjunto de parámetros de secuencia, es posible codificar los elementos de un RPS (por ejemplo, el RPS 400) con referencia a otra RPS ya codificada en el conjunto de parámetros de secuencia. Esto se denomina Inter RPS. No hay problemas de robustez de error asociados con este método ya que todos los RPS del conjunto de parámetros de secuencia están en la misma unidad de capa de abstracción de red (NAL). La sintaxis Inter RPS aprovecha el hecho de que la RPS de la imagen actual puede predecirse a partir del RPS de una imagen decodificada previamente. Esto se debe a que todas las imágenes de referencia de la imagen actual deben ser imágenes de referencia de la imagen anterior o de la propia imagen decodificada previamente. Sólo es necesario indicar cuáles de estas imágenes deben ser imágenes de referencia y utilizarse para la predicción de la imagen actual. Por lo tanto, la sintaxis comprende lo siguiente: un índice que apunta a la RPS que se utilizará como predictor, un `delta_POC` que se añadirá al `delta_POC` del predictor para obtener el `delta_POC` del RPS actual, y un conjunto de indicadores para indicar qué imágenes son imágenes de referencia y si sólo se utilizan para la predicción de imágenes futuras. En una realización, el `delta_POC` se refiere a la diferencia en el valor POC entre una imagen de referencia actual y otra (por ejemplo, anterior).

65

Los codificadores que deseen explotar 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 pueden señalarse en la cabecera de fragmento mediante palabras clave de longitud fija, `poc_lsb_lt`, que representan los bits menos significativos del valor POC completo de cada imagen a largo plazo. Cada

5 `poc_lsb_lt` es una copia de la palabra clave `pic_order_cnt_lsb` señalada para una imagen a largo plazo concreta. También es posible señalar un conjunto de imágenes de largo plazo en el SPS como una lista de valores POC LSB. El POC LSB de una imagen de largo plazo puede señalizarse en la cabecera de fragmento como un índice de esta lista.

10 El elemento sintáctico `delta_poc_msb_cycle_lt_minus1` puede señalizarse adicionalmente para permitir el cálculo de la distancia POC completa de una imagen de referencia a largo plazo en relación con la imagen actual. Se requiere que la palabra clave `delta_poc_msb_cycle_lt_minus1` se señalice para cada imagen de referencia a largo plazo que tenga el mismo valor LSB de POC que cualquier otra imagen de referencia en el RPS.

15 Para el marcado de imágenes de referencia en HEVC, normalmente habrá varias imágenes presentes en la DPB antes de la decodificación de la imagen. Algunas de ellas pueden estar disponibles para la predicción y, por tanto, marcadas como "usadas para referencia". Otras pueden no estar disponibles para la predicción, pero estar a la espera de la salida, por lo que se marcan como "no utilizadas para referencia". Una vez analizada la

20 cabecera de fragmento, se lleva a cabo un proceso de marcado de imágenes antes de decodificar los datos del fragmento. Las imágenes presentes en la DPB y marcadas como "usadas para referencia" pero no incluidas en el RPS se marcan como "no usadas para referencia". Las imágenes que no están presentes en la DPB, pero están incluidas en el conjunto de imágenes de referencia se ignoran si el indicador `used_by_curr_pic_x_flag` es igual a cero. Sin embargo, si el indicador `used_by_curr_pic_x_flag` es igual a uno, esta imagen de referencia debía utilizarse para la predicción en la imagen actual, pero no está. En ese caso,

25 se infiere una pérdida de imagen involuntaria y el decodificador debe tomar las medidas oportunas.

Tras decodificar la imagen actual, se marca como "utilizada como referencia a corto plazo".

30 A continuación, se analiza la construcción de listas de imágenes de referencia en HEVC. En HEVC, el término interpredicción se utiliza para designar la predicción derivada de elementos de datos (por ejemplo, valores de muestra o vectores de movimiento) de imágenes de referencia distintas de la imagen decodificada actual. Como en AVC, una imagen puede predecirse a partir de varias imágenes de referencia. Las imágenes de referencia que se utilizan para la interpredicción se organizan en una o más listas de imágenes de referencia.

35 El índice de referencia identifica cuál de las imágenes de referencia de la lista debe utilizarse para crear la señal de predicción.

Se utiliza una única lista de imágenes de referencia, la Lista 0, para un fragmento P y dos listas de imágenes de referencia, la Lista 0 y la Lista 1, para los fragmentos B. Al igual que en AVC, la construcción de la lista de

40 imágenes de referencia en HEVC incluye la inicialización y la modificación de la lista de imágenes de referencia.

En AVC, el proceso de inicialización de la Lista 0 es diferente para los fragmentos P (para los que se utiliza el orden de decodificación) y los fragmentos B (para los que se utiliza el orden de salida). En HEVC, el orden de salida se utiliza en ambos casos.

45 La inicialización de la lista de imágenes de referencia crea por defecto la Lista 0 y la Lista 1 (si el fragmento es un fragmento B) basándose en tres subconjuntos RPS: `RefPicSetStCurrBefore`, `RefPicSetStCurrAfter` y `RefPicSetLtCurr`. Las imágenes a corto plazo con orden de salida anterior (posterior) se insertan primero en la Lista 0 (Lista 1) en orden ascendente de distancia POC a la imagen actual, después las imágenes a corto plazo con orden de salida posterior (anterior) se insertan en la Lista 0 (Lista 1) en orden ascendente de distancia

50 POC a la imagen actual y, por último, las imágenes a largo plazo se insertan al final. En términos de RPS, para la Lista 0, las entradas de `RefPicSetStCurrBefore` se insertan en la lista inicial, seguidas de las entradas de `RefPicSetStCurrAfter`. A continuación, se añaden las entradas de `RefPicSetLtCurr`, si están disponibles.

55 En HEVC, el proceso anterior se repite (las imágenes de referencia que ya se han añadido a la lista de imágenes de referencia se vuelven a añadir) cuando el número de entradas de una lista es inferior al número objetivo de imágenes de referencia activas (indicado en el conjunto de parámetros de imagen o en la cabecera de fragmento). Cuando el número de entradas es superior al número objetivo, la lista se trunca.

60 Una vez inicializada la lista de imágenes de referencia, puede modificarse de modo que las imágenes de referencia de la imagen actual se dispongan en cualquier orden, incluyendo el caso en que una imagen de referencia concreta pueda aparecer en más de una posición de la lista, con base en los comandos de modificación de la lista de imágenes de referencia. Cuando la bandera que indica la presencia de modificaciones en la lista se establece en uno, se señala un número fijo (igual al número objetivo de entradas

65 en la lista de imágenes de referencia) de comandos, y cada comando inserta una entrada para una lista de

imágenes de referencia. Una imagen de referencia se identifica en el comando por el índice de la lista de imágenes de referencia para la imagen actual derivada de la señalización RPS. Esto difiere de la modificación de la lista de imágenes de referencia en H.264/AVC, en la que una imagen se identifica mediante el número de imagen (derivado del elemento sintáctico `frame_num`) o el índice de imagen de referencia a largo plazo, y es posible que se necesiten menos comandos, por ejemplo, para intercambiar las dos primeras entradas de una lista inicial o insertar una entrada al principio de la lista inicial y desplazar las demás.

Una lista de imágenes de referencia no puede incluir ninguna imagen de referencia con un `TemporalId` superior al de la imagen actual. Un flujo de bits HEVC puede constar de varias subcapas temporales. Cada unidad NAL pertenece a una subcapa específica indicada 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 la gestión de las imágenes de referencia en la DPB, evitando así la necesidad de los procesos de señalización y decodificación para 1) la ventana deslizante y los procesos MMCO, así como los procesos de inicialización y modificación de la lista de imágenes de referencia en AVC, o 2) el conjunto de imágenes de referencia, así como los procesos de inicialización y modificación de la lista de imágenes de referencia en HEVC.

Por desgracia, la gestión de imágenes de referencia basada en listas de imágenes de referencia (RPL) presenta inconvenientes. Por ejemplo, algunos esquemas de gestión de imágenes de referencia basados en RPL no están optimizados en términos de señalización de elementos sintácticos en el flujo de bits. Esto hace que estos enfoques basados en RPL tengan un mayor número de bits para la señalización en comparación con otros enfoques de gestión explícita de imágenes de referencia, tal como un enfoque basado en RPS. La ineficacia de señalización de los enfoques con base en RPL se debe a varios aspectos.

Por ejemplo, algunos elementos sintácticos de una estructura RPL se codifican con una codificación de entropía menos eficiente. Por ejemplo, la codificación del elemento de sintaxis que representa el valor delta POC de las imágenes de referencia a corto plazo (STRPs) utiliza un elemento de sintaxis codificado con Exp-Golomb de orden 0-vo entero con signo con el bit de la izquierda primero (por ejemplo, `se(v)`) porque el valor delta POC puede tener un valor positivo o negativo. Para codificar cualquier número entero no negativo x utilizando el código exp-Golomb, el primer paso es escribir $x+1$ en binario. A continuación, se cuentan los bits escritos, se resta uno y se escribe ese número de bits de cero inicial precediendo a la cadena de bits anterior. Los primeros 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, esto puede hacerse más eficiente si el valor del signo y el valor absoluto del delta POC se codifican por separado.

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

Además, no hay codificación inter-RPL para escribir una estructura de lista de imágenes de referencia en un flujo de bits de vídeo codificado. Por ejemplo, las estructuras RPL pueden codificarse sin referencia a la(s) estructura(s) RPL previamente señalada(s) en el flujo de bits. En una realización, una estructura RPL se refiere a una construcción de programación que contiene una lista indexada de punteros a imágenes de referencia candidatas.

En el presente documento se describen varios aspectos, que pueden aplicarse individualmente y/o en combinación, que resuelven los problemas o abordan los inconvenientes descritos en el presente documento. Cada uno de estos aspectos se detalla a continuación.

En un aspecto, el valor de signo y el valor absoluto se codifican por separado en el flujo de bits de vídeo cuando los valores delta POC se codifican para STRPs. Para indicar si los valores delta POC en cada estructura RPL tienen el mismo valor de signo (por ejemplo, si todos los delta POC en la misma estructura RPL son valores delta POC positivos o valores delta POC negativos), se señala una bandera en el mismo conjunto de parámetros (por ejemplo, SPS, PPS) que las estructuras RPL predefinidas. El indicador puede denominarse "`all_rpl_entries_same_sign_flag`".

Cuando la bandera `all_rpl_entries_same_sign_flag` es igual a 1, se señala una bandera en la estructura RPL para indicar el valor del signo de todas las entradas (por ejemplo, el valor del signo de los valores delta POC de los STRP) en la estructura RPL. Cuando la bandera `all_rpl_entries_same_sign_flag` es igual a 0, se señala

una bandera en la estructura RPL para indicar el valor del signo de cada una de las entradas asociadas con STRP en cada estructura RPL. En una realización, el valor absoluto de un valor delta POC de las entradas STRP en la estructura RPL se codifica utilizando una codificación Exp-Golomb de 0-ésimo orden entero sin signo (por ejemplo, ue(v)).

5

Alternativamente, en lugar de una bandera para la bandera `all_rpl_entries_same_sign_flag`, se pueden señalar dos banderas; una para cada lista de estructuras RPL (por ejemplo, lista de estructuras RPL para RPL0 y lista de estructuras RPL para RPL1).

10

Para la codificación, al crear estructuras RPL predefinidas para las 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 forma que la estructura RPL para la lista 0 y la estructura RPL para la lista 1 estén emparejadas. De este modo, cuando una imagen hace referencia a las estructuras RPL predefinidas en el conjunto de parámetros, los índices para RPL0 y RPL0 de dicha imagen son los mismos.

15

En relación con el punto anterior, una bandera puede indicar si los elementos de sintaxis para RPL 1 que hacen referencia a la estructura predefinida RPL están presentes o no en una cabecera de fragmento. La bandera puede denominarse "`rpl1_idx_present_flag`". Esta bandera puede ser señalizada en SPS o PPS dependiendo del alcance previsto o persistencia de la bandera. En una realización, se prefiere la señalización de la bandera en PPS.

20

Con base en el valor de `rpl1_idx_present_flag`, se aplica lo siguiente. Cuando la `rpl1_idx_present_flag` es igual a 1, si el RPL 1 de la cabecera de fragmento asociado con el conjunto de parámetros que contiene el indicador se refiere a la estructura RPL predefinida y su índice no se indica en la cabecera de fragmento y, en su lugar, se deduce 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 se señalan, sino que se deducen (por ejemplo, se copian) del valor de `ref_pic_list_sps_flag[0]` y `ref_pic_list_idx[0]`, respectivamente. De lo contrario, 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 fragmento asociada con el conjunto de parámetros que contiene la bandera.

25

30

Alternativamente, con base en el valor de `rpl1_idx_present_flag`, se aplica lo siguiente. Cuando la `rpl1_idx_present_flag` es igual a 1, el índice de RPL 1 no se indica en la cabecera de fragmento cuando RPL 1 de la cabecera de fragmento asociada con el conjunto de parámetros que contiene la bandera hace referencia a la estructura RPL predefinida. En su lugar, el índice de RPL 1 se deduce del elemento sintáctico correspondiente para RPL 0 en el mismo fragmento. Es decir, `ref_pic_list_idx[1]` no se señala, sino que se infiere (por ejemplo, se copia) del valor de `ref_pic_list_idx[0]`, respectivamente. De lo contrario, cuando la `rpl1_idx_present_flag` es igual a 0, `ref_pic_list_idx[1]` está presente en la cabecera de fragmento asociada con el conjunto de parámetros que contiene la bandera.

35

40

Puede utilizarse una bandera para señalar si las estructuras RPL de la lista para RPL 1 tienen idéntico contenido que las estructuras RPL de la lista para RPL 0. La bandera puede denominarse "`rpl1_copy_from_rpl0_flag`". La bandera se señaliza en el mismo conjunto de parámetros que contiene las estructuras RPL predefinidas y se situará antes de la señalización de las estructuras RPL predefinidas.

45

Con base en el valor de `rpl1_copy_from_rpl0_flag`, se aplica lo siguiente. Cuando el indicador `rpl1_copy_from_rpl0_flag` es igual a 1, el número de estructuras RPL de la lista 1 no se señaliza y, en su lugar, se deduce que es el mismo que el número de estructuras RPL de la lista 0. No se señaliza ninguna estructura RPL de la lista 1. En su lugar, una vez analizadas y decodificadas en el decodificador las estructuras RPL predefinidas para la lista 0, para cada estructura RPL de la lista 0 se crea una copia idéntica de la estructura RPL, y la copia de la estructura RPL se asigna como estructura RPL de la lista 1 con el mismo índice. En caso contrario, cuando la `rpl1_copy_from_rpl0_flag` es igual a 0, se señaliza el número de estructuras RPL para la lista 1 y de estructuras RPL para la lista 1.

50

El contenido de una estructura RPL puede predecirse a partir de otra estructura RPL. Cuando las 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), sólo las estructuras RPL de la lista para la lista 0 pueden utilizarse como referencia para inter-RPL. Una estructura RPL de la lista para la lista 0 sólo puede referirse a otra estructura RPL de la misma lista con un índice inferior a su índice, mientras que una estructura RPL de la lista para la lista 1 puede referirse a cualquier estructura RPL de la lista para la lista 0. La estructura RPL señalada explícitamente en la cabecera de fragmento puede referirse a cualquiera de las estructuras RPL predefinidas de la lista para la lista 0. Alternativamente, una estructura RPL de la lista para la lista 0 o la lista 1 sólo puede referirse a otra estructura RPL de la lista para la lista 0 con un índice inferior a su índice. La estructura RPL señalada explícitamente en la cabecera de fragmento puede hacer referencia a cualquiera de las estructuras RPL predefinidas en la lista para la lista 0.

60

65

En una alternativa, cuando las 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 la lista 0 sólo puede referirse a otra estructura RPL en la lista para la lista 0 con un índice menor que su índice. Del mismo modo, una estructura RPL de la lista para la lista 1 sólo puede referirse a otra estructura RPL de la lista para la lista 1 con un índice menor que su índice. La estructura RPL señalada explícitamente en la cabecera de fragmento para RPL0 puede referirse a cualquiera de las estructuras RPL predefinidas en la lista para la lista 0 y la estructura RPL señalada explícitamente en la cabecera de fragmento para RPL 1 puede referirse a cualquiera de las estructuras RPL predefinidas en la lista para la lista 1.

En otra alternativa, cuando las estructuras RPL predefinidas en un conjunto de parámetros no están divididas en dos listas (por ejemplo, la lista de estructuras RPL para la lista 0 y la lista de estructuras RPL para la lista 1 están señaladas en una lista), una estructura RPL sólo puede hacer referencia a otra estructura RPL de la lista con un índice menor que su índice. La estructura RPL señalada explícitamente en la cabecera de fragmento puede referirse a cualquiera de las estructuras RPL predefinidas.

Para inter-RPL, el índice de la estructura RPL de referencia puede codificarse como el delta del índice de la estructura RPL actual y el índice de la estructura RPL de referencia menos 1 y codificarse con la codificación $ue(v)$. En una alternativa, el índice RPL de referencia se codifica directamente utilizando codificación $u(v)$. El número de bits utilizados para representar el índice se establece como \log_2 del número de estructuras RPL de la lista elegibles para referencia. Por ejemplo, cuando sólo las estructuras RPL de la lista para la lista 0 pueden utilizarse como referencia, el número de bits para representar el índice RPL de referencia es \log_2 del número de estructuras RPL de la lista para la lista 0. En otra alternativa, el índice para la estructura RPL de referencia puede codificarse utilizando $ue(v)$ o $u(v)$ dependiendo del modo de inter-RPL.

Para soportar inter-RPL, cada estructura RPL se codifica utilizando uno de los modos descritos a continuación. Nótese que el orden del modo a continuación no significa necesariamente el orden para el valor del modo. La indicación del modo puede codificarse como sigue. En una realización, la indicación del modo puede codificarse simplemente utilizando $ue(v)$. En una realización, la indicación del modo puede codificarse utilizando $u(v)$ con el número de bits para representar el modo siendo \log_2 del número total de los modos definidos.

El primer modo de codificación RPL es un modo de intracodificación. Este modo funciona igual que los otros mecanismos para señalar el contenido de la estructura RPL. Véase, por ejemplo, el método descrito en Solicitud Provisional de Estados Unidos No. 62/719,360 presentada el 17 de agosto de 2018 y titulada "Reference Picture Management in Video Coding". Alternativamente, puede señalarse un valor entero mayor que 0 para cada estructura RPL y denominarse $granularity_val$. El valor de $granularity_val$ se utiliza para escalar o dividir cada valor que representa los valores delta de POC de STRPs en la estructura RPL.

El segundo modo de codificación RPL es un modo de intercodificación que utiliza los valores delta POC para STRPs 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, se señala la siguiente información en el flujo de bits.

Para empezar, el modo de codificación y el índice de referencia se señalan en el flujo de bits. El índice de referencia es el índice de la estructura RPL de referencia. El índice de referencia puede codificarse como se ha descrito anteriormente. Sin embargo, los mecanismos también pueden codificar un índice de referencia como el delta entre el índice de la estructura RPL actual y el índice de la estructura RPL de referencia menos 1 y codificado como $ue(v)$. El desplazamiento también se indica en el flujo de bits. El desplazamiento es la diferencia entre el delta POC del STRP en la estructura RPL de referencia y el delta POC del STRP en la estructura RPL actual. El valor de desplazamiento puede limitarse a ser sólo positivo (por ejemplo, si el valor delta POC del STRP en la estructura RPL de referencia es menor que el valor delta POC del STRP en la estructura RPL actual, este modo no puede utilizarse para codificar la estructura RPL actual), sólo negativo, o puede ser tanto positivo como negativo. El desplazamiento puede señalarse con $term_minus1$ si se codifica como $ue(v)$. También se indica en el flujo de bits una bandera para cada entrada de la estructura RPL de referencia que indica si la entrada se utiliza como entrada en la estructura RPL actual. Cuando una entrada es una entrada STRP en la estructura RPL de referencia y se utiliza como 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 la entrada en la estructura RPL de referencia menos el desplazamiento (más 1 si se utiliza el término $minus1$ para codificar el desplazamiento). Cuando una entrada es una entrada de imagen de referencia a largo plazo (LTRP) en la estructura RPL de referencia y se utiliza como entrada en la estructura RPL actual, la entrada también es una entrada LTRP en la estructura RPL actual y su valor simplemente se copia para la entrada en la estructura RPL de referencia.

El tercer modo de codificación RPL es un modo de intercodificación que utiliza el hecho de que las entradas (por ejemplo, STRP y LTRP) en la estructura RPL actual son un superconjunto de entradas en la estructura

RPL de referencia. Es decir, todas las entradas de la estructura RPL de referencia son las mismas que la primera X (sustituya X por el número de entradas de la estructura RPL de referencia) con cero o más entradas adicionales. Para codificar RPL utilizando este modo, se señala la siguiente información en el flujo de bits.

- 5 Para empezar, el modo de codificación y el índice de referencia se señalan en el flujo de bits. El índice de referencia es el índice de la estructura RPL de referencia. El índice de referencia puede codificarse como se ha descrito anteriormente. El número de entradas adicionales también se indica en el flujo de bits. El número de entradas de la estructura RPL de referencia es la diferencia entre el número de entradas de la estructura RPL actual y el número de entradas de la estructura RPL de referencia. Cuando una entrada es una entrada STRP en la estructura RPL de referencia, la entrada también es una entrada STRP en la estructura RPL actual y su valor simplemente se copia 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 simplemente se copia para la entrada en la estructura RPL de referencia. Una vez copiadas todas las entradas de la estructura RPL de referencia en la estructura RPL actual, por cada entrada adicional se señala la siguiente información: Si se utiliza una imagen de referencia a largo plazo en el flujo de bits (es decir, esto puede indicarse mediante una bandera en el mismo conjunto de parámetros), se señala una bandera para indicar si la entrada adicional es una entrada LTRP o una entrada STRP. Si se trata de una entrada LTRP, se indica el POC LSB de la entrada LTRP. En caso contrario, se señala el POC delta de la entrada STRP. El valor del delta POC puede señalarse como el delta de la entrada STRP anterior o simplemente el delta del POC de la imagen actual.

El cuarto modo de codificación RPL es un modo de intercodificación que utiliza el hecho de que las entradas (por ejemplo, STRP y LTRP) en la estructura RPL actual son exactamente las mismas que las entradas en la estructura RPL de referencia o exactamente las mismas con el valor del signo invertido. Para codificar los RPL utilizando este modo, se señala la siguiente información en el flujo de bits.

Para empezar, se señalan el modo de codificación y el índice de referencia. El índice de referencia es el índice de la estructura RPL de referencia. El índice de referencia puede codificarse como se ha descrito anteriormente. Opcionalmente, también se señala en el flujo de bits una bandera para indicar si se ha invertido el valor del signo.

Cuando un flujo de bits se codifica con interpredicción hacia delante (por ejemplo, interpredicción con referencia a una imagen con un valor de POC inferior al valor de POC de la imagen actual) e interpredicción hacia atrás (por ejemplo, interpredicción con referencia a una imagen con un valor de POC superior al valor de POC de la imagen actual), se aplican las siguientes restricciones a las imágenes de referencia de un RPL. Para cada RPL de una imagen, todas las imágenes de referencia en el RPL son de la misma dirección de interpredicción, por ejemplo, todas las imágenes de referencia son imágenes de referencia para la interpredicción hacia delante o todas las imágenes de referencia son imágenes de referencia para la interpredicción hacia atrás. Para el par de RPL de una imagen, no habrá ninguna imagen de referencia contenida tanto en RPL0 como en RPL1 a menos que se cumpla la siguiente condición. La condición es que tanto RPL0 como RPL1 contengan las mismas imágenes de referencia con el mismo orden. En tal caso, todas las imágenes de referencia, excepto la primera (es decir, la de menor índice) de RPL1, se eliminan de RPL1. Alternativamente, todas las imágenes de referencia, excepto la primera (es decir, la de menor índice) en el RPL0, se eliminan del RPL0.

En el presente documento se divulgan técnicas de codificación de vídeo que utilizan una bandera para especificar si se puede deducir que un índice de una segunda estructura de lista de imágenes de referencia es el mismo que un índice de una primera estructura de lista de imágenes de referencia. Es decir, un índice a la segunda estructura de lista de imágenes de referencia no está presente en una cabecera de fragmento del flujo de bits de vídeo codificado y se infiere que es el mismo que un índice a la primera estructura de lista de imágenes de referencia cuando la bandera tiene un primer valor. Por otra parte, el índice de la segunda estructura de lista de imágenes de referencia está presente en la cabecera de fragmento cuando la bandera tiene un segundo valor. Al utilizar la bandera de esta manera, el codificador / decodificador (también conocido como "códec") en la codificación de vídeo se mejora (por ejemplo, utiliza menos bits, exige menos ancho de banda, es más eficiente, etc.) en relación con los códecs actuales. En la práctica, el proceso de codificación de vídeo mejorado ofrece al usuario una mejor experiencia cuando envía, recibe y/o visualiza vídeos.

La FIG. 5 es un diagrama esquemático de una realización de un flujo 500 de bits de vídeo. En el presente documento, el flujo 500 de bits de vídeo también puede denominarse flujo de bits de vídeo codificado, flujo de bits o variaciones del mismo. Como se muestra en la FIG. 5, el flujo 500 de bits comprende un conjunto de parámetros de secuencia (SPS) 510, un conjunto de parámetros de imagen (PPS) 512, una cabecera 514 de fragmento, y datos 520 de imagen.

El SPS 510 contiene datos comunes a todas las imágenes de una secuencia de imágenes (SOP). Por el contrario, el PPS 512 contiene datos comunes a toda la imagen. La cabecera 514 de fragmento contiene información sobre el fragmento actual tal como, por ejemplo, el tipo de fragmento, cuál de las imágenes de

referencia se utilizará, etcétera. El SPS 510 y el PPS 512 pueden referirse genéricamente como un conjunto de parámetros. El SPS 510, el PPS 512 y la cabecera 514 de fragmento son tipos de unidades de la capa de abstracción de red (NAL). Los datos 520 de imagen comprenden datos asociados con las imágenes o vídeo que están siendo codificados o decodificados. Los datos 520 de imagen pueden denominarse simplemente
5 carga útil o datos transportados en el flujo 500 de bits.

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

La FIG. 6 es una realización de un método 600 de decodificación de un flujo de bits de vídeo codificado (por ejemplo, flujo 500 de bits) implementado por un decodificador de vídeo (por ejemplo, decodificador 30 de vídeo). El método 600 puede realizarse después de que el flujo de bits codificado se haya recibido directa o indirectamente de un codificador de vídeo (por ejemplo, el codificador 20 de vídeo). El método 600 mejora el
15 proceso de decodificación (por ejemplo, hace que el proceso de decodificación sea más eficiente, más rápido, etc., que los procesos de decodificación convencionales) porque se puede inferir que un índice de una segunda estructura de lista de imágenes de referencia es el mismo que un índice de la primera estructura de lista de imágenes de referencia cuando una bandera se establece en un valor determinado. Es decir, no es necesario
20 señalar la segunda estructura de lista de imágenes de referencia en el flujo de bits codificado en todas las circunstancias, como en HEVC y AVC. Por lo tanto, en la práctica, el rendimiento de un códec mejora, lo que se traduce en una mejor experiencia para el usuario.

En el bloque 602, se analiza un indicador del flujo de bits de vídeo codificado (por ejemplo, el flujo 500 de bits de vídeo). En una realización, el indicador se designa `rpl1_idx_present_flag`. En una realización, el indicador se incluye en el PPS (por ejemplo, PPS 512) del flujo de bits de vídeo codificado. En una realización, el
25 indicador se incluye en el SPS (por ejemplo, SPS 510) del flujo de bits de vídeo codificado.

En una realización, el primer valor de la bandera es uno (1). En una realización, una `ref_pic_list_sps_flag[1]` y una `ref_pic_list_idx[1]` no se incluyen en la cabecera de fragmento cuando el primer valor de la marca es uno (1). En una realización, el segundo valor de la bandera es cero (0). En una realización, una
30 `ref_pic_list_sps_flag[0]` y una `ref_pic_list_idx[0]` se incluyen en la cabecera de fragmento cuando el segundo valor de la bandera es cero (0).

En el bloque 604, se analiza una primera estructura de lista de imágenes de referencia a partir del flujo de bits de vídeo codificado. En una realización, la primera estructura de lista de imágenes de referencia se incluye en la cabecera de fragmento (por ejemplo, la cabecera 514 de fragmento) del flujo de bits de vídeo codificado. En una realización, la bandera y la primera estructura de lista de imágenes de referencia se analizan
35 separadamente una de otra. Es decir, primero se analiza la bandera y luego la primera estructura de lista de imágenes de referencia, o viceversa.
40

En el bloque 606, se determina que un índice de una segunda estructura de lista de imágenes de referencia no está presente en una cabecera de fragmento del flujo de bits de vídeo codificado y se infiere que el índice de la segunda estructura de lista de imágenes de referencia es el mismo que un índice de la primera estructura de lista de imágenes de referencia cuando el indicador tiene un primer valor. En el bloque 608, se determina
45 que el índice de la segunda estructura de lista de imágenes de referencia está presente en la cabecera de fragmento cuando la bandera tiene un segundo valor.

En el bloque 610, se genera una lista de imágenes de referencia utilizando la primera estructura de lista de imágenes de referencia, 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 tales como, por ejemplo, las imágenes ilustradas y descritas en relación con la FIG. 4.
50

En el bloque 612, se realiza una interpretación con base en la lista de imágenes de referencia para generar un bloque reconstruido. En una realización, el bloque reconstruido puede utilizarse para generar o producir una imagen que se mostrará a un usuario en la pantalla de un dispositivo electrónico (por ejemplo, un teléfono inteligente, una tableta, un ordenador portátil, un ordenador personal, etc.).
55

La FIG. 7 es una realización de un método 700 de codificación de un flujo de bits de vídeo (por ejemplo, flujo 500 de bits) implementado por un codificador de vídeo (por ejemplo, codificador 20 de vídeo). El método 700 puede ser realizado cuando una imagen (por ejemplo, de un vídeo) debe ser codificada en un flujo de bits de vídeo y luego transmitida hacia un decodificador de vídeo (por ejemplo, decodificador 30 de vídeo). 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) porque se puede inferir que un índice de una segunda estructura de lista de imágenes de referencia es el mismo que un índice de la primera estructura
60
65

de lista de imágenes de referencia cuando una bandera se establece en un valor determinado. Es decir, no es necesario señalar la segunda estructura de lista de imágenes de referencia en el flujo de bits codificado en todas las circunstancias, como en HEVC y AVC. Por lo tanto, en la práctica, el rendimiento de un códec mejora, lo que se traduce en una mejor experiencia para el usuario.

5

En el bloque 702, se codifica una bandera con un primer valor en el flujo de bits de vídeo cuando un índice de una segunda estructura de lista de imágenes de referencia no está codificado en una cabecera de fragmento del flujo de bits de vídeo y un decodificador de vídeo debería deducir que el índice de la segunda estructura de lista de imágenes de referencia es el mismo que el índice de una primera estructura de lista de imágenes de referencia. En una realización, el indicador se designa como `rpl1_idx_present_flag`. En una realización, el indicador está codificado en el PPS (por ejemplo, PPS 512) del flujo de bits de vídeo codificado (por ejemplo, flujo 500 de bits de vídeo). En una realización, la primera estructura de lista de imágenes de referencia se codifica en la cabecera de fragmento (por ejemplo, cabecera 514 de fragmento) del flujo de bits de vídeo.

10

En el bloque 704, una bandera con un segundo valor se codifica en el flujo de bits de vídeo cuando el índice de la segunda estructura de lista de imágenes de referencia se codifica en la cabecera de fragmento del flujo de bits de vídeo. En una realización, la primera estructura de lista de imágenes de referencia y la segunda lista de imágenes de referencia se codifican en la cabecera de fragmento del flujo de bits de vídeo codificado.

15

En el bloque 706, la primera estructura de lista de imágenes de referencia se codifica en el flujo de bits de vídeo cuando la bandera se codifica 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 se codifican en el flujo de bits de vídeo cuando la bandera se codifica con el segundo valor. En una realización, el primer valor de la bandera es uno (1) y el segundo valor de la bandera es cero (0). En una realización, una `ref_pic_list_sps_flag[1]` y una `ref_pic_list_idx[1]` no se incluyen en la cabecera de fragmento cuando el primer valor de la bandera es uno (1).

20

25

En el bloque 710, el flujo de bits de vídeo se transmite hacia el decodificador de vídeo (por ejemplo, el decodificador 30 de vídeo). Una vez recibido por el decodificador de vídeo, el flujo de bits de vídeo codificado puede ser decodificado (por ejemplo, como se ha descrito anteriormente) para generar o producir una imagen para mostrar a un usuario en la pantalla de un dispositivo electrónico (por ejemplo, un teléfono inteligente, tableta, ordenador portátil, ordenador personal, etc.).

30

Se proporciona una descripción de las técnicas divulgadas en el presente documento en relación con el último WC WD. Además, se proporcionan definiciones, sintaxis y semántica adecuadas para implementar las técnicas descritas en el presente documento.

35

Para empezar, se ofrecen varias definiciones. Una imagen de punto de acceso intraaleatorio (IRAP) es una imagen codificada para la que cada unidad NAL de la capa de codificación de vídeo (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 que cada unidad NAL de la VCL tiene `nal_unit_type` igual a `NON_IRAP_NUT`. Una lista de imágenes de referencia es una lista de imágenes de referencia utilizadas para la interpredicción de un fragmento P o B. Se generan dos listas de imágenes de referencia, la lista de imágenes de referencia 0 y la lista de imágenes de referencia 1, para cada sección de una imagen no IRAP.

40

45

El conjunto de imágenes únicas a las que hacen referencia todas las entradas de las dos listas de imágenes de referencia asociadas a una imagen incluye todas las imágenes de referencia que pueden utilizarse para la interpredicción de la imagen asociada o de cualquier imagen que siga a la imagen asociada en el orden de decodificación. Para decodificar los datos de fragmento de un fragmento P, sólo se utiliza la lista de imágenes de referencia 0 para la interpredicción. Para decodificar los datos de fragmento de un fragmento B, se utilizan ambas listas de imágenes de referencia para la interpredicción. Para decodificar los datos de fragmento de un fragmento I, no se utiliza ninguna lista de imágenes de referencia para la interpredicción. Una imagen de referencia a corto plazo (STRP) es una imagen marcada como "utilizada para referencia a corto plazo".

50

55

A continuación, se indican algunas abreviaturas. En el presente documento, LTRP son las siglas de Long-Term Reference Picture (imagen de referencia a largo plazo) y STRP son las siglas de Short-Term Reference Picture (imagen de referencia a corto plazo).

60

Las siguientes porciones de la divulgación proporcionan la sintaxis y la semántica de acuerdo con la invención reivindicada.

Sintaxis de la cabecera de la unidad nal

<code>nal_unit_header() {</code>	Descriptor
----------------------------------	-------------------

ES 2 935 979 T3

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

Conjunto de parámetros de secuencia Sintaxis RBSP

	Descriptor
seq_parameter_set_rbsp() {	
sps_seq_parameter_set_id	ue(v)
chroma_format_idc	ue(v)
if(chroma_format_idc == 3)	
separate_colour_plane_flag	u(1)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
bit_depth_luma_minus8	ue(v)
bit_depth_chroma_minus8	ue(v)
log2_max_pic_order_cnt_lsb_minus4	ue(v)
sps_max_dec_pic_buffering_minus1	ue(v)
qtbtt_dual_tree_intra_flag	ue(v)
log2_ctu_size_minus2	ue(v)
log2_min_qt_size_intra_slices_minus2	ue(v)
log2_min_qt_size_inter_slices_minus2	ue(v)
max_mtt_hierarchy_depth_inter_slices	ue(v)
max_mtt_hierarchy_depth_intra_slices	ue(v)
long_term_ref_pics_flag	u(1)
if(long_term_ref_pics_flag)	
additional_lt_poc_lsb	ue(v)
all_rpl_entries_same_sign_flag	u(1)
rpl1_copy_from_rp10_flag	u(1)
for(i = 0; i < (!rpl1_copy_from_rp10_flag) ? 2 : 1; i++) {	
num_ref_pic_lists_in_sps[i]	ue(v)
for(j = 0; j < num_ref_pic_lists_in_sps[i]; j++)	
ref_pic_list_struct(i, j, long_term_ref_pics_flag)	
}	
rbsp_trailing_bits()	
}	

Conjunto de parámetros de imagen Sintaxis RBSP

	Descriptor
pic_parameter_set_rbsp() {	
pps_pic_parameter_set_id	ue(v)
pps_seq_parameter_set_id	ue(v)
for(i = 0; i < 2; i++)	

num_ref_idx_default_active_minus1[i]	ue(v)
rpl1_idx_present_flag	u(1)
rbsp_trailing_bits()	
}	

Sintaxis de la cabecera de fragmento

	Descriptor
slice_header() {	
slice_pic_parameter_set_id	ue(v)
slice_address	u(v)
slice_type	ue(v)
if (slice_type != I)	
log2_diff_ctu_max_bt_size	ue(v)
if(nal_unit_type != IRAP_NUT) {	
slice_pic_order_cnt_lsb	u(v)
for(i = 0; i < 2; i++) {	
if(i == 0 (i == 1 && rpl1_idx_present_flag))	
ref_pic_list_sps_flag[i]	u(1)
if(ref_pic_list_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1)	
if(i == 0 (i == 1 && rpl1_idx_present_flag))	
ref_pic_list_idx[i]	u(v)
} else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i], long_term_ref_pics_flag)	
}	
if(slice_type == P slice_type == B) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag)	
for(i = 0; i < (slice_type == B ? 2: 1); i++)	
num_ref_idx_active_minus1[i]	ue(v)
}	
}	
byte_alignment()	
}	

Sintaxis de la estructura de la lista de imágenes de referencia

	Descriptor
ref_pic_list_struct(listIdx, rplIdx, ltrpFlag) {	
rpl_mode[listIdx][rplIdx]	u(2)
if(rpl_mode[listIdx][rplIdx] == 0) {	
num_strp_entries[listIdx][rplIdx]	ue(v)
if(ltrpFlag)	
num_ltrp_entries[listIdx][rplIdx]	ue(v)
if(all_rpl_entries_same_sign_flag)	

strp_entries_sign_flag [listIdx][rplsIdx]	u(1)
for(i = 0; i < NumEntriesInList[listIdx][rplsIdx]; i++) {	
if(num_itr_entries[listIdx][rplsIdx] > 0)	
lt_ref_pic_flag [listIdx][rplsIdx][i]	u(1)
if(!lt_ref_pic_flag[listIdx][rplsIdx][i]) {	
if(!all_rpl_entries_same_sign_flag)	
strp_entry_sign_flag [listIdx][rplsIdx][i]	u(1)
delta_poc_st [listIdx][rplsIdx][i]	ue(v)
} else	
poc_lsb_lt [listIdx][rplsIdx][i]	u(v)
}	
} else if(rpl_mode[listIdx][rplsIdx] == 1) {	
model_ref_rpl_idx_delta_minus1 [listIdx][rplsIdx]	ue(v)
strp_offset_val_minus1 [listIdx][rplsIdx]	ue(v)
for(i = 0; i < NumEntriesInList[0][RefRplIdx]; i++)	
ref_entry_used_flag [listIdx][rplsIdx][i]	u(1)
} else if(rpl_mode[listIdx][rplsIdx] == 2) {	
mode2_ref_rpl_idx [listIdx][rplsIdx]	u(v)
num_additional_entries [listIdx][rplsIdx]	ue(v)
for(i = 0; i < num_additional_entries[listIdx][rplsIdx]; i++) {	
if(!trpFlag)	
add_lt_ref_pic_flag [listIdx][rplsIdx][i]	u(1)
if(!add_lt_ref_pic_flag[listIdx][rplsIdx][i]) {	
if(!all_rpl_entries_same_sign_flag)	
add_strp_entry_sign_flag [listIdx][rplsIdx][i]	u(1)
add_delta_poc_st [listIdx][rplsIdx][i]	ue(v)
} else	
add_poc_lsb_lt [listIdx][rplsIdx][i]	u(v)
}	
} else if(rpl_mode[listIdx][rplsIdx] == 3)	
mode3_ref_rpt_idx [listIdx][rplsIdx]	u(v)
}	

Semántica de la cabecera de unidad NAL

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

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 la unidad NAL y estructura sintáctica de la RBSP	Clase de tipo de unidad NAL
0	NON_IRAP_NUT	Segmento codificado de una imagen no IRAP slice_layer_rbsp()	VCL

Nal_unit_type	Nombre de nal_unit_type	Contenido de la unidad NAL y estructura sintáctica de la RBSP	Clase de tipo de unidad NAL
1	IRAP_NUT	Fragmento 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 del flujo de bits end_of_bitstream_rbsp()	no VCL
20,21	PREFIJO_SEI_NUEZ SUFIJO_SEI_NUEZ	Información complementaria de mejora sei_rbsp()	no VCL
22-26	RSV_NVCL	Reservado	no VCL
27-31	UNSPEC	Sin especificar	no VCL

La variable nuh_temporal_id_plus1 menos 1 especifica un identificador temporal para la unidad NAL. El valor de nuh_temporal_id_plus1 no será igual a 0. La variable TemporalId se especifica como sigue: TemporalId = nuh_temporal_id_plus1 - 1. Cuando nal_unit_type es igual a IRAP_NUT, el fragmento 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 de una unidad de acceso es el valor de TemporalId de las unidades VCL NAL de la imagen codificada o de la unidad de acceso. El valor de TemporalId de las unidades NAL no VCL se limita del siguiente modo: Si nal_unit_type es igual a SPS_NUT, TemporalId será igual a 0 y el TemporalId de la unidad de acceso que contenga la unidad NAL será igual a 0. En caso contrario, si nal_unit_type es igual a EOS_NUT o EOB_NUT, TemporalId será igual a 0. En caso contrario, TemporalId será mayor o igual que el TemporalId de la unidad de acceso que contenga la unidad NAL. Cuando la unidad NAL es una unidad NAL no VCL, el valor de TemporalId es igual al valor mínimo de los valores de TemporalId de todas las unidades de acceso a las que se aplica la unidad NAL no VCL. Cuando nal_unit_type es igual a PPS_NUT, TemporalId puede ser mayor o igual que TemporalId de la unidad de acceso que contiene, ya que todos los conjuntos de parámetros de imagen (PPS) pueden incluirse al principio de un flujo de bits, en el que la primera imagen codificada tiene TemporalId igual a 0. Cuando nal_unit_type es igual a PREFIJO_SEI_NUEZ o SUFIJO_SEI_NUEZ, TemporalId puede ser mayor o igual que TemporalId de la unidad de acceso que contiene la unidad SEI NAL, ya que una unidad SEI NAL puede contener información que se aplica a un subconjunto de flujo de bits que incluye unidades de acceso cuyos valores de TemporalId son mayores que 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 podrán ser especificados en el futuro por el UIT-T | ISO/IEC. Los decodificadores ignorarán (es decir, eliminarán del flujo de bits y descartarán) las unidades NAL con valores de nuh_reserved_zero_7bits distintos de "0000000".

25 Conjunto de parámetros de secuencia Semántica de la 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 el recuento del orden de las imágenes de la siguiente manera: $MaxPicOrderCntLsb = 2^{(log2_max_pic_order_cnt_lsb_minus4 + 4)}$. El valor de log2_max_pic_order_cnt_lsb_minus4 deberá estar comprendido entre 0 y 12, ambos inclusive. sps_max_dec_pic_buffering_minus1 plus 1 especifica el tamaño máximo requerido de la memoria intermedia de imágenes decodificadas para la secuencia de vídeo codificada (CVS) en unidades de memoria intermedia de imágenes. El valor de sps_max_dec_pic_buffering_minus1 estará comprendido entre 0 y MaxDpbSize - 1, ambos inclusive, donde MaxDpbSize es el especificado en otro lugar. long_term_ref_pics_flag igual a 0 especifica que no se utiliza LTRP para la interpredicción de ninguna imagen codificada en la CVS. long_term_ref_pics_flag igual a 1 especifica que pueden utilizarse LTRP para la interpredicción 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 las listas de imágenes de referencia de la siguiente manera: $MaxLtPicOrderCntLsb = 2^{(log2_max_pic_order_cnt_lsb_minus4 + 4 + additional_lt_poc_lsb)}$. El valor de additional_lt_poc_lsb estará comprendido entre 0 y 32 - log2_max_pic_order_cnt_lsb_minus4 - 4, ambos inclusive. Si no está presente, se deduce que el valor de additional_lt_poc_lsb es igual a 0.

Una `all_rpl_entries_same_sign_flag` igual a 1 especifica que todas las entradas STRP de 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 las entradas STRP de cada `ref_pic_list_struct(listIdx, rplIdx, ltrpFlag)` pueden tener o no 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]`. Se infiere que la estructura sintáctica `ref_pic_list_struct(1, rplIdx, ltrpFlag)` es idéntica al valor de `ref_pic_list_struct(0, rplIdx, ltrpFlag)`. En consecuencia, se deduce que los elementos sintácticos de `ref_pic_list_struct(1, rplIdx, ltrpFlag)` son respectivamente iguales a los elementos sintácticos de `ref_pic_list_struct(0, rplIdx, ltrpFlag)`.

Un `num_ref_pic_lists_in_sps[i]` especifica el número de estructuras de sintaxis `ref_pic_list_struct(listIdx, rplIdx, ltrpFlag)` con `listIdx` igual a `i` incluidas en el SPS. El valor de `num_ref_pic_lists_in_sps[i]` estará comprendido entre 0 y 64, ambos inclusive. Para cada valor de `listIdx` (igual a 0 o 1), un decodificador deberá asignar memoria para un número total de estructuras de sintaxis `num_ref_pic_lists_in_sps[i] + 1` `ref_piclist_struct(listIdx, rplIdx, ltrpFlag)`, ya que puede haber una estructura de sintaxis `ref_pic_list_struct(listIdx, rplIdx, ltrpFlag)` señalada directamente en las cabeceras de fragmento de una imagen actual.

20 Conjunto de parámetros de imagen Semántica de la RBSP

Un `num_ref_idx_default_active_minus1[i]` más 1, cuando `i` es igual a 0, especifica el valor inferido de la variable `NumRefIdxActive[0]` para fragmentos 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 fragmentos B con `num_ref_idx_active_override_flag` igual a 0. El valor de `num_ref_idx_default_active_minus1 [i]` estará en el intervalo de 0 a 14, ambos 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 fragmento. `rpl1_idx_present_flag` igual a 1 especifica que `ref_pic_list_sps_flag[1]` y `ref_pic_list_idx[1]` pueden estar presentes en las cabeceras de fragmento.

30 Semántica de la cabecera de fragmento

Cuando está presente, el valor de cada uno de los elementos de sintaxis de cabecera de fragmento `slice_pic_parameter_set_id` y `slice_pic_order_cnt_lsb` deberá ser el mismo en todas las cabeceras de fragmento de una imagen codificada.... `slice_type` especifica el tipo de codificación del fragmento de acuerdo con la Tabla 7-3.

Tabla 7-3 - Asociación del nombre al `slice_type`

tipo de corte	Nombre del <code>slice_type</code>
0	B (fragmento B)
1	P (fragmento P)
2	I (fragmento I)

40 Cuando `nal_unit_type` es igual a IRAP NUT, es decir, la imagen es una imagen IRAP, `slice_type` será igual a 2.... `slice_pic_order_cnt_lsb` especifica el recuento de orden de imagen módulo `MaxPicOrderCntLsb` para la imagen actual. La longitud del elemento de sintaxis `slice_pic_order_cnt_lsb` es $\log_2(\text{max_pic_order_cnt_lsb_minus4} + 4)$ bits. El valor de `slice_pic_order_cnt_lsb` estará comprendido entre 0 y `MaxPicOrderCntLsb - 1`, inclusive. Cuando `slice_pic_order_cnt_lsb` no está presente, se infiere que `slice_pic_order_cnt_lsb` es igual a 0. `ref_pic_list_sps_flag[i]` igual a 1 especifica que la lista de imágenes de referencia `i` de la imagen actual se deriva con base en una de las estructuras de sintaxis `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 de imágenes de referencia `i` de la imagen actual se deriva con base en la estructura sintáctica `ref_pic_list_struct(listIdx, rplIdx, ltrpFlag)` con `listIdx` igual a `i` que está incluida directamente en las cabeceras de fragmento 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, se infiere que el valor de `ref_pic_list_sps_flag[1]` es igual al valor de `ref_piclist_sps_flag[0]`. `ref_pic_list_idx[i]` especifica el índice, en la lista de las estructuras sintácticas `ref_piclist_struct(listIdx, rplIdx, ltrpFlag)` con `listIdx` igual a `i` incluidas en el SPS activo, de la estructura sintáctica `ref_pic_list_struct(listIdx, rplIdx, ltrpFlag)` con `listIdx` igual a `i` que se utiliza para la derivación de la lista de imágenes de referencia `i` de la imagen actual. El elemento sintáctico `ref_pic_list_idx[i]` está representado por bits $\text{Ceil}(\log_2(\text{num_ref_pic_lists_in_sps}[i]))$. Cuando no está presente, se deduce que el valor de `ref_pic_list_idx[i]` es igual a 0. El valor de `ref_pic_list_idx[i]` estará comprendido entre 0 y `num_ref_piclists_in_sps[i] - 1`,

inclusive. Cuando `rpl1_idx_present` flag es igual a 0 y `ref_pic_list_sps_flag[0]` está presente, se infiere que el valor de `ref_pic_list_idx[1]` es igual al valor de `ref_pic_list_idx[0]`. `num_ref_idx_active_override_flag` igual a 1 especifica que el elemento sintáctico `num_ref_idx_active_minus1[0]` está presente para los fragmentos P y B y que el elemento sintáctico `num_ref_idx_active_minus1[1]` está presente para los fragmentos B.

5 `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]` como sigue: `NumRefIdxActive[i] = num_ref_idx_active_minus1[i] + 1`. El valor de `num_ref_idx_active_minus1[i]` estará comprendido entre 0 y 14, inclusive.

10 El valor de `NumRefIdxActive[i] - 1` especifica el índice de referencia máximo para la lista de imágenes de referencia `i` que puede utilizarse para decodificar el corte. Cuando el valor de `NumRefIdxActive[i]` es igual a 0, no se puede utilizar ningún índice de referencia para la lista de imágenes de referencia `i` para decodificar el corte. Para `i` igual a 0 ó 1, cuando la sección actual es una sección B y `num_ref_idx_active_override_flag` es igual a 0, se deduce que `NumRefIdxActive[i]` es igual a `num_ref_idx_default_active_minus1[i] + 1`. Cuando la sección actual es una sección P, no se puede utilizar ningún índice de referencia para la lista de imágenes de referencia `i` para decodificar la sección. Cuando la sección actual es una sección P y `num_ref_idx_active_override_flag` es igual a 0, se deduce que `NumRefIdxActive[0]` es igual a `num_ref_idx_default_active_minus1[0] + 1`. Cuando la sección actual es una sección P, se infiere que

15 `NumRefIdxActive[1]` es igual a 0. Cuando la sección actual es una sección I, se infiere que tanto `NumRefIdxActive[0]` como `NumRefIdxActive[1]` son iguales a 0.

Alternativamente, para `i` igual a 0 o 1, se aplica lo siguiente después de lo anterior: Sea `rplsIdx1` igual a `ref_piclist_sps_flag[i] ? ref_pic_list_idx[i] : num_ref_pic_lists_in_sps[i]`, y `numRpEntries[i]` igual a `num_strp_entries[i][rplsIdx1] + num_ltrp_entries[i][rplsIdx1]`. Cuando `NumRefIdxActive[i]` es mayor que `numRpEntries[i]`, el valor de `NumRefIdxActive[i]` se establece igual a `numRpEntries[i]`.

25

Semántica de la estructura de la lista de imágenes de referencia

30 La estructura sintáctica `ref_pic_list_struct(listIdx, rplsIdx, ltrpFlag)` puede estar presente en un SPS o en una cabecera de fragmento. Dependiendo de si la estructura sintáctica está incluida en una cabecera de fragmento o en un SPS, se aplica lo siguiente: Si está presente en una cabecera de fragmento, la estructura sintáctica `ref_pic_list_struct(listIdx, rplsIdx, ltrpFlag)` especifica la lista de imágenes de referencia `listIdx` de la imagen actual (la imagen que contiene el fragmento). En caso contrario (presente en un SPS), la estructura sintáctica

35 `ref_pic_list_struct(listIdx, rplsIdx, 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 fragmentos que contienen `ref_pic_list_idx[listIdx]` igual a un índice en la lista de las estructuras sintácticas `ref_pic_list_struct(listIdx, rplsIdx, ltrpFlag)` incluidas en el SPS, y 2) está en un CVS que tiene el SPS como SPS activo. `rpl_mode[listIdx][rplsIdx]` especifica el modo de codificación de los elementos de sintaxis en la estructura de sintaxis `ref_pic_list_struct(listIdx, rplsIdx, ltrpFlag)`. `num_strp_entries[listIdx][rplsIdx]` especifica el número de entradas STRP en la estructura de sintaxis `ref_pic_list_struct(listIdx, rplsIdx, ltrpFlag)`. `num_ltrp_entries[listIdx][rplsIdx]` especifica el número de entradas LTRP en la estructura de sintaxis `ref_pic_list_struct(listIdx, rplsIdx, ltrpFlag)`. Cuando no está presente, se infiere que el valor de `num_ltrp_entries[listIdx][rplsIdx]` es igual a 0. La variable `NumEntriesInList[listIdx][rplsIdx]` se obtiene como sigue: `NumEntriesInList[listIdx][rplsIdx] = num_strp_entries[listIdx][rplsIdx] + num_ltrp_entries[listIdx][rplsIdx]`. El valor de `NumEntriesInList[listIdx][rplsIdx]` estará en el intervalo de 0 a `sps_max_dec_pic_buffering_minus1`, inclusive. `strp_entries_sign_flag[listIdx][rplsIdx]` igual a 1 especifica que todas las entradas STRP en `ref_pic_list_struct(listIdx, rplsIdx, ltrpFlag)` tienen un valor mayor o igual a 0. `strp_entries_sign_flag[listIdx][rplsIdx]` igual a 0 especifica que todas las entradas STRP en

40 `ref_pic_list_struct(listIdx, rplsIdx, ltrpFlag)` tienen un valor menor que 0.

45

`lt_ref_pic_flag[listIdx][rplsIdx][i]` igual a 1 especifica que la entrada `i`-ésima de la estructura sintáctica `ref_piclist_struct(listIdx, rplsIdx, ltrpFlag)` es una entrada LTRP. `lt_ref_pic_flag[listIdx][rplsIdx][i]` igual a 0 especifica que la entrada `i`-ésima de la estructura sintáctica `ref_pic_list_struct(listIdx, rplsIdx, ltrpFlag)` es una entrada STRP. Cuando no está presente, se infiere que el valor de `lt_ref_pic_flag[listIdx][rplsIdx][i]` es igual a 0. Es un requisito de conformidad del flujo de bits que la suma de `lt_ref_pic_flag[listIdx][rplsIdx][i]` para todos los valores de `i` en el intervalo de 0 a `NumEntriesInList[listIdx][rplsIdx] - 1`, inclusive, sea igual a `num_ltrp_entries[listIdx][rplsIdx]`. `strp_entry_sign_flag[listIdx][rplsIdx][i]` igual a 1 especifica que la entrada `i`-ésima de `ref_pic_list_struct(listIdx, rplsIdx, ltrpFlag)` tiene un valor mayor o igual a 0. `strp_entries_sign_flag[listIdx][rplsIdx]` igual a 0 especifica que la `i`-ésima entrada en `ref_pic_list_struct(listIdx, rplsIdx, ltrpFlag)` tiene un valor menor que 0. Cuando no está presente, se infiere que el valor de `strp_entries_sign_flag[listIdx][rplsIdx][i]` es igual al valor de `strp_entries_sign_flag[listIdx][rplsIdx]`.

55

60

El valor `delta_poc_st[listIdx][rplsIdx][i]`, cuando la entrada `i`-ésima es la primera entrada STRP en la estructura sintáctica `ref_pic_list_struct(rplsIdx, ltrpFlag)`, especifica la diferencia entre los valores de recuento de orden

65

de imágenes de la imagen actual y la imagen a la que hace referencia la entrada i-ésima, o, cuando la entrada i-ésima es una entrada STRP pero no la primera entrada STRP de la estructura sintáctica ref_pic_list_struct(rplsIdx, ltrpFlag), especifica la diferencia entre los valores de recuento de orden de imágenes de las imágenes a las que hace referencia la entrada i-ésima y la entrada STRP anterior de la estructura sintáctica ref_pic_list_struct(listIdx, rplsIdx, ltrpFlag). El valor de delta_poc_st[listIdx][rplsIdx][i] estará comprendido entre -215 y 215 - 1, inclusive. poc_lsb_lt[listIdx][rplsIdx][i] especifica el valor del recuento de orden de imagen módulo MaxLtPicOrderCntLsb de la imagen a la que se refiere la entrada i-ésima de la estructura sintáctica ref_pic_list_struct(listIdx, rplsIdx, ltrpFlag). La longitud del elemento sintáctico poc_lsb_lt[listIdx][rplsIdx][i] es $\text{Log}_2(\text{MaxLtPicOrderCntLsb})$ bits. La matriz DeltaPocSt[listIdx][rplsIdx] se obtiene como sigue:

```

5
10
15
20
    for(i = 0; i < NumEntriesInList[ listIdx ][ rplsIdx ]; i++) {
        if(!lt_ref_pic_flag[ i ][ RplsIdx[ i ] ][ j ]) {
            DeltaPocSt[ listIdx ][ rplsIdx ][ i ] = (strp_entry_sign_flag[ listIdx ][ rplsIdx ][ i ] ) ?
                delta_poc_st[ listIdx ][ rplsIdx ][ i ] : 0 - delta_poc_st[ listIdx ][ rplsIdx ][ i ]
        }
    }

```

El model1_ref_rpl_idx_delta_minus1[listIdx][rplsIdx] más 1 especifica la diferencia entre el valor de rplsIdx y el índice de la referencia ref_pic_list_struct(listIdx, rplsIdx, ltrpFlag). Cuando rpl_mode[listIdx][rplsIdx] es igual a 1, la variable RefRplIdx se deriva como sigue:

```

25
    RefRplIdx = rplsIdx - ( model1_ref_rpl_idx_delta_minus1[ listIdx ][ rplsIdx ] + 1 )

```

30 El strp_offset_val_minus1[listIdx][rplsIdx] más 1 especifica el valor que se restará de cada entrada STRP de la ref_pic_list_struct(listIdx, rplsIdx, ltrpFlag) para calcular el valor de delta_poc_st[listIdx][rplsIdx][i] de la ref_pic_list_struct(listIdx, rplsIdx, ltrpFlag) actual. La ref_entry_used_flag[listIdx][rplsIdx][i] igual a 1 especifica que la entrada i-ésima de ref_pic_list_struct(0, RefRplIdx, ltrpFlag) se utiliza como entrada en ref_pic_list_struct(listIdx, rplsIdx, ltrpFlag). Ref_entry_used_flag[listIdx][rplsIdx] igual a 0 especifica que la entrada i-ésima de ref_pic_list_struct(0, RefRplIdx, ltrpFlag) no se utiliza como entrada en ref_pic_list_struct(listIdx, rplsIdx, ltrpFlag).

40 Cuando rpl_mode[listIdx][rplsIdx] es igual a 1, se aplica lo siguiente para la inferencia de los valores de los elementos sintácticos 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 la 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]:

```

45
50
55
60
65
    currEntryIdx = 0
    for(i = 0; i < NumEntriesInList[ 0 ][ RefRplsIdx ]; i++) {
        if(ref_entry_used_flag[ listIdx ][ rplsIdx ][ i ]) {
            lt_ref_pic_flag[ listIdx ][ rplsIdx ][ currEntryIdx ] = lt_ref_pic_flag[ 0 ][ RefRplsIdx ][ i ]
        }
        if(!lt_ref_pic_flag[ 0 ][ RefRplsIdx ][ i ])
            DeltaPocSt[ hstIdx ][ rplsIdx ][ currEntryIdx ] = DeltaPocSt[ 0 ][ RefRplsIdx ][ i ]
        else
            poc_lsb_lt[ listIdx ][ rplsIdx ][ currEntryIdx ] = poc_lsb_lt[ 0 ][ RefRplsIdx ][ i ]
        currEntryIdx++
    }
    NumEntriesInList[ listIdx ][ rplsIdx ] = currEntryIdx

```

El elemento `mode2_ref_rpl_idx[listIdx][rplsIdx]` especifica el índice de la referencia `ref_pic_list_struct(listIdx, rplsIdx, ltrpFlag)`. El elemento sintáctico `mode2_ref_rpl_idx[listIdx][rplsIdx]` está representado por bits $\text{Ceil}(\text{Log}_2(\text{num_ref_pic_lists_in_sps}[0]))$. Cuando `rpl_mode[listIdx][rplsIdx]` es igual a 2, la variable `RefRplIdx` se obtiene como sigue:

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

El `num_additional_entries[listIdx][rplsIdx]` especifica el delta entre `NumEntriesInList[listIdx][rplsIdx]` y `NumEntriesInList[0][RefRplIdx]`. `add_lt_ref_pic_flag[listIdx][rplsIdx][i]` se utiliza para inferir el valor de `lt_ref_pic_flag[listIdx][rplsIdx][NumEntriesInList[0][RefRplIdx] + i]`. Si no está presente, se infiere que el valor de `add_lt_ref_pic_flag[listIdx][rplsIdx][i]` es igual a 0. El valor de `add_strp_entry_sign_flag[listIdx][rplsIdx][i]` se utiliza para inferir el valor de `strp_entry_sign_flag[listIdx][rplsIdx][NumEntriesInList[0][RefRplIdx] + i]`. Cuando no está presente, el valor de `add_strp_entries_sign_flag[listIdx][rplsIdx][i]` se establece igual al valor de `strp_entries_sign_flag[listIdx][rplsIdx]`.

El `add_delta_poc_st[listIdx][rplsIdx][i]` se utiliza para inferir el valor de `delta_poc_st[listIdx][rplsIdx][NumEntriesInList[0][RefRplIdx] + i]`. El valor de `add_delta_poc_st[listIdx][rplsIdx][i]` estará comprendido entre -215 y 215-1, ç inclusive. `add_poc_lsb_lt[listIdx][rplsIdx][i]` se utiliza para inferir el 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 $\text{Log}_2(\text{MaxLtPicOrderCntLsb})$ 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 `strp_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 la 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]`:

```

30   strp_entries_sign_flag[ listIdx ][ rplsIdx ] = strp_entries_sign_flag[ 0 ][ RefRplIdx ]
    for(i = 0; i < NumEntriesInList[ 0 ][ RefRplIdx ]; i++) {
        lt_ref_pic_flag[ listIdx ][ rplsIdx ][ i ] = lt_ref_pic_flag[ 0 ][ RefRplIdx ][ i ]
35   if(!lt_ref_pic_flag[ 0 ][ RefRplIdx ][ i ])
        DeltaPocSt[ listIdx ][ rplsIdx ][ i ] = DeltaPocSt[ 0 ][ RefRplIdx ][ i ]
    else
40   poc_lsb_lt[ listIdx ][ rplsIdx ][ i ] = poc_lsb_lt[ 0 ][ RefRplIdx ][ i ]
    }
45   currEntryIdx = NumEntriesInList[ 0 ][ RefRplIdx ]
    for(i = 0; i < num_additional_entries[ listIdx ][ rplsIdx ]; i++, currEntryIdx++) {
        lt_ref_pic_flag[ listIdx ][ rplsIdx ][ currEntryIdx ] = add_lt_ref_pic_flag[ listIdx ][ rplsIdx ][ i ]
50   if(!add_lt_ref_pic_flag[ listIdx ][ rplsIdx ][ i ])
        DeltaPocSt[ listIdx ][ rplsIdx ][ currEntryIdx ] = (add_strp_entry_sign_flag[ listIdx ][ rplsIdx ][ i ] ?
55   add_delta_poc_st[ listIdx ][ rplsIdx ][ i ] : 0 - add_delta_poc_st[ listIdx ][ rplsIdx ][ i ]
    else
        poc_lsb_lt[ listIdx ][ rplsIdx ][ currEntryIdx ] = add_poc_lsb_lt[ listIdx ][ rplsIdx ][ i ]
60   }
    NumEntriesInList[ listIdx ][ rplsIdx ] = currEntryIdx

```

El `mode3_ref_rpl_idx[listIdx][rplsIdx]` especifica el índice de la referencia `ref_pic_list_struct(listIdx, rplsIdx,`

ltrpFlag). El elemento sintáctico mode3_ref_rpl_idx[listIdx][rplsIdx] está representado por bits $\lceil \log_2(\text{num_ref_pic_lists_in_sps}[0]) \rceil$. Cuando rpl_mode[listIdx][rplsIdx] es igual a 3, la variable RefRplIdx se deriva como sigue:

5 RefRplIdx = mode3_ref_rpl_idx[listIdx][rplsIdx]

Cuando rpl_mode[listIdx][rplsIdx] es igual a 3, 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_isb_lt[listIdx][rplsIdx][i], y para la 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]:

```

10     for(i = 0; i < NumEntriesInList[ 0 ][ RefRplsIdx ]; i++) {
15         lt_ref_pic_flag[ listIdx ][ rplsIdx ][ i ] = lt_ref_pic_flag[ 0 ][ RefRplsIdx ][ i ]
16         if(!lt_ref_pic_flag[ 0 ][ RefRplsIdx ][ i ])
17             DeltaPocSt[ listIdx ][ rplsIdx ][ i ] = 0 - DeltaPocSt[ 0 ][ RefRplsIdx ][ i ]
20     else
21         poc_isb_lt[ listIdx ][ rplsIdx ][ i ] = poc_isb_lt[ 0 ][ RefRplsIdx ][ i ]
22     }
25     NumEntriesInList[ listIdx ][ rplsIdx ] = NumEntriesInList[ 0 ][ RefRplsIdx ]

```

Se proporciona un proceso general de decodificación.

30 El proceso de decodificación funciona del siguiente modo para la imagen actual CurrPic: La decodificación de unidades NAL se especifica a continuación. Los procesos que se indican a continuación especifican los siguientes procesos de decodificación utilizando elementos de sintaxis de la capa de cabecera de fragmento y superiores: Se derivan variables y funciones relacionadas con el recuento de orden de imagen. Esto sólo se invoca para el primer fragmento de una imagen. Al principio del proceso de decodificación de cada segmento de una imagen no IRAP, se invoca el proceso de decodificación para la construcción de listas de imágenes de referencia para la derivación de la lista de imágenes de referencia 0 (RefPicList[0]) y la lista de imágenes de referencia 1 (RefPicList[1]). Se invoca el proceso de decodificación para el marcado de imágenes de referencia, en el que las imágenes de referencia pueden marcarse como "no utilizadas para referencia" o "utilizadas para referencia a largo plazo". Esto se invoca sólo para el primer fragmento de una imagen. Se invocan los procesos de decodificación para unidades de árbol de codificación, escalado, transformación, filtrado en bucle, etc. Una vez decodificados todos los fragmentos de la imagen actual, ésta se marca como "utilizada para referencia a corto plazo".

45 Se proporciona un proceso de decodificación de unidades NAL.

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

50 Se proporciona un proceso de decodificación de fragmentos.

El proceso de decodificación para el recuento del orden de las imágenes es el siguiente.

55 La salida de este proceso es PicOrderCntVal, el recuento de orden de imagen de la imagen actual. Los recuentos de orden de imagen se utilizan para identificar imágenes, para derivar parámetros de movimiento en modo de fusión y predicción de vector de movimiento, y para la comprobación de conformidad del decodificador. Cada imagen codificada está asociada con una variable de recuento de orden de imagen, denominada PicOrderCntVal. Cuando la imagen actual no es una imagen IRAP, las variables prevPicOrderCntLsb y prevPicOrderCntMsb se derivan como sigue: Sea prevTid0Pic 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 es igual a PicOrderCntMsb de prevTid0Pic.

65 La variable PicOrderCntMsb de la imagen actual se obtiene del siguiente modo: Si la imagen actual es una

imagen IRAP, PicOrderCntMsb se establece igual a 0. En caso contrario, PicOrderCntMsb se obtiene del siguiente modo:

```

5  if((slice_pic_order_cnt_lsb < prevPicOrderCntLsb) &&
    ((prevPicOrderCntLsb - slice_pic_order_cnt_lsb) >= (MaxPicOrderCntLsb / 2)))
    PicOrderCntMsb = prevPicOrderCntMsb + MaxPicOrderCntLsb
10 else if((slice_pic_order_cnt_lsb > prevPicOrderCntLsb) &&
    ((slice_pic_order_cnt_lsb - prevPicOrderCntLsb) > (MaxPicOrderCntLsb / 2)))
    PicOrderCntMsb = prevPicOrderCntMsb - MaxPicOrderCntLsb
15 else
    PicOrderCntMsb = prevPicOrderCntMsb
20 PicOrderCntVal se obtiene de la siguiente manera: PicOrderCntVal = PicOrderCntMsb +
    slice_pic_order_cnt_lsb.

```

25 Todas las imágenes IRAP tendrán PicOrderCntVal igual a 0, ya que se deduce que slice_pic_order_cnt_lsb es 0 para las imágenes IRAP y prevPicOrderCntLsb y prevPicOrderCntMsb son iguales a 0. El valor de PicOrderCntVal estará comprendido entre -231 y 231 - 1, ambos inclusive. En un CVS, los valores PicOrderCntVal de dos imágenes codificadas cualesquiera no deberán ser iguales.

30 En cualquier momento del proceso de decodificación, los valores de PicOrderCntVal & (MaxLtPicOrderCntLsb - 1) de dos imágenes de referencia cualesquiera de la DPB no deberán coincidir. La función PicOrderCnt(picX) se especifica como sigue: PicOrderCnt(picX) = PicOrderCntVal de la imagen picX. La función DiffPicOrderCnt(picA, picB) se especifica como sigue: DiffPicOrderCnt(picA, picB) = PicOrderCnt(picA) - PicOrderCnt(picB). El flujo de bits no contendrá datos que den lugar a valores de DiffPicOrderCnt(picA, picB) utilizados en el proceso de decodificación que no estén comprendidos entre -215 y 215 - 1, ambos inclusive. Si X es la imagen actual e Y y Z son otras dos imágenes del mismo CVS, se considerará que Y y Z están en la misma dirección de orden de salida que X cuando tanto DiffPicOrderCnt(X, Y) como DiffPicOrderCnt(X, Z) sean positivos o ambos negativos.

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

40 Este proceso se invoca al principio del proceso de decodificación para cada corte de una imagen no IRAP. Las imágenes de referencia se direccionan mediante índices de referencia. Un índice de referencia es un índice de una lista de imágenes de referencia. Cuando se decodifica un fragmento I, no se utiliza ninguna lista de imágenes de referencia en la decodificación de los datos del fragmento. Cuando se decodifica un fragmento P, sólo se utiliza la lista de imágenes de referencia 0 (es decir, RefPicList[0]) en la decodificación de los datos del fragmento. Cuando se decodifica un fragmento B, tanto la lista de imágenes de referencia 0 como la lista de imágenes de referencia 1 (es decir, RefPicList[1]) se utilizan en la decodificación de los datos del fragmento. Al principio del proceso de decodificación de cada fragmento de una imagen no IRAP, se obtienen las listas de imágenes de referencia RefPicList[0] y RefPicList[1]. Las listas de imágenes de referencia se utilizan para marcar las imágenes de referencia o para decodificar los datos de los fragmentos. En el caso de un fragmento I de una imagen no IRAP que no sea el primer fragmento de la imagen, RefPicList[0] y RefPicList[1] pueden derivarse con fines de comprobación de la conformidad del flujo de bits, pero su derivación no es necesaria para decodificar la imagen actual o las imágenes que siguen a la imagen actual en el orden de decodificación. En el caso de un fragmento P que no sea el primer fragmento de una imagen, RefPicList[1] puede derivarse con fines de comprobación de la conformidad del flujo de bits, pero su derivación no es necesaria para la decodificación de la imagen o imágenes actuales que siguen a la imagen actual en el orden de decodificación. Las listas de imágenes de referencia RefPicList[0] y RefPicList[1] se construyen como sigue:

```

    for(i = 0; i < 2; i++) {
60  if(_ref_pic_list_sps_flag[ i ])
    RplIdx[ i ] = ref_pic_list_idx[ i ]
    else
65

```

```

RplsIdx[ i ] = num_ref_pic_lists_in_sps[ i ]
for(j = 0, pocBase = PicOrderCntVal; j < NumEntriesInList[ i ][ RplsIdx[ i ] ]; j++) {
5   if(!lt_ref_pic_flag[ i ][ RplsIdx[ i ] ][ j ]) {
      RefPicPocList[ i ][ j ] = pocBase - DeltaPocSt[ i ][ RplsIdx[ i ] ][ j ]
10   if(there is a reference picture picA in the DPB with PicOrderCntVal equal to RefPicPocList[ i ][ j ])
      RefPicList[ i ][ j ] = picA
      else
15   RefPicList[ i ][ j ] = "no reference picture"
      pocBase = RefPicPocList[ i ][ j ]
   } else {
20   if(there is a reference picA in the DPB with PicOrderCntVal & (MaxLtPicOrderCntLsb - 1)
      equal a poc_lsb_lt[ i ][ RplsIdx[ i ] ][ j ])
25   RefPicList[ i ][ j ] = picA
      else
30   RefPicList[ i ][ j ] = "no reference picture"
      }
      }
35 }

```

Para cada i igual a 0 o 1, se aplica lo siguiente: Las primeras entradas de $\text{NumRefIdxActive}[i]$ en $\text{RefPicList}[i]$ se denominan entradas activas de $\text{RefPicList}[i]$, y las demás entradas de $\text{RefPicList}[i]$ se denominan entradas inactivas de $\text{RefPicList}[i]$. Cada entrada de $\text{RefPicList}[i][j]$ para j en el intervalo de 0 a $\text{NumEntriesInList}[i][\text{RplsIdx}[i]] - 1$, inclusive, se denomina entrada STRP si $\text{lt_ref_pic_flag}[i][\text{RplsIdx}[i]][j]$ es igual a 0, y entrada LTRP en caso contrario. Es posible que se haga referencia a una imagen concreta particular en una entrada de $\text{RefPicList}[0]$ como en una entrada de $\text{RefPicList}[1]$. También es posible que se haga referencia a una imagen concreta en más de una entrada de $\text{RefPicList}[0]$ o en más de una entrada de $\text{RefPicList}[1]$. Las entradas activas en $\text{RefPicList}[0]$ y las entradas activas en $\text{RefPicList}[1]$ se refieren colectivamente a todas las imágenes de referencia que pueden utilizarse para la interpretación 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 $\text{RefPicList}[1]$ se refieren colectivamente a todas las imágenes de referencia que no se utilizan para la interpretación de la imagen actual pero que pueden utilizarse en la interpretación de 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 "ninguna imagen de referencia" porque las imágenes correspondientes no están presentes en la DPB. Cada entrada inactiva en $\text{RefPicList}[0]$ o $\text{RefPicList}[1]$ que sea igual a "sin imagen de referencia" debe ignorarse. Debe inferirse una pérdida de imagen involuntaria para cada entrada activa en $\text{RefPicList}[0]$ o $\text{RefPicList}[1]$ que sea igual a "sin imagen de referencia".

Es un requisito de conformidad del flujo de bits que se apliquen las siguientes restricciones: Para cada i igual a 0 o 1, $\text{NumEntriesInList}[i][\text{RplsIdx}[i]]$ no será inferior a $\text{NumRefIdxActive}[i]$. La imagen a la que se refiere cada entrada activa en $\text{RefPicList}[0]$ o $\text{RefPicList}[1]$ deberá estar presente en el DPB y deberá tener TemporalId menor o igual que la de la imagen actual. Opcionalmente, puede especificarse además la siguiente restricción: El índice de entrada de cualquier entrada inactiva en $\text{RefPicList}[0]$ o $\text{RefPicList}[1]$ no se utilizará como índice de referencia para decodificar la imagen actual. Opcionalmente, puede especificarse la siguiente restricción: Una entrada inactiva en $\text{RefPicList}[0]$ o $\text{RefPicList}[1]$ no se referirá a la misma imagen que cualquier otra entrada en $\text{RefPicList}[0]$ o $\text{RefPicList}[1]$. Una entrada STRP en $\text{RefPicList}[0]$ o $\text{RefPicList}[1]$ de un fragmento de una imagen y una entrada LTRP en $\text{RefPicList}[0]$ o $\text{RefPicList}[1]$ de la misma sección o de una sección diferente de la misma imagen no harán referencia a la misma imagen. Ninguna entrada de

RefPicList[0] o RefPicList[1] hará referencia a la propia imagen actual. No habrá ninguna entrada LTRP en RefPicList[0] o RefPicList[1] para la que la diferencia entre el PicOrderCntVal de la imagen actual y el PicOrderCntVal de la imagen a la que hace referencia la entrada sea mayor o igual que 224. Sea setOfRefPics el conjunto de imágenes únicas a las que hacen referencia todas las entradas de RefPicList[0] y todas las entradas de RefPicList[1]. El número de imágenes en setOfRefPics deberá ser menor o igual que $sps_max_dec_pic_buffering_minus1$ y setOfRefPics deberá ser el mismo para todos los fragmentos de una imagen.

Se proporciona un proceso de decodificación para el marcado de imágenes de referencia.

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

La FIG. 8 es un diagrama esquemático de un dispositivo 800 de codificación de vídeo (por ejemplo, un codificador 20 de vídeo o un decodificador 30 de vídeo) de acuerdo con una realización de la divulgación. El dispositivo 800 de codificación de vídeo es adecuado para implementar las realizaciones divulgadas tal como se describen en el presente documento. El dispositivo 800 de codificación de vídeo comprende puertos 810 de entrada y unidades receptoras (Rx) 820 para recibir datos; un procesador, unidad lógica o unidad central de procesamiento (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 vídeo también puede comprender componentes óptico-eléctricos (OE) y componentes eléctrico-ópticos (EO) acoplados a los puertos 810 de entrada, las unidades 820 receptoras, las unidades 840 transmisoras, y los puertos 850 de salida para la salida o entrada de señales ópticas o eléctricas.

El procesador 830 se implementa mediante hardware y software. El procesador 830 puede implementarse como uno o más chips de CPU, núcleos (por ejemplo, como un procesador multinúcleo), matrices de puertas programables en campo (FPGA), circuitos integrados de aplicación específica (ASIC) y procesadores de señales digitales (DSP). El procesador 830 está en comunicación con los puertos 810 de entrada, las unidades 820 receptoras, las unidades 840 transmisoras, los puertos 850 de salida y la memoria 860. El procesador 830 comprende un módulo 870 de codificación. El módulo 870 de codificación implementa las realizaciones divulgadas descritas anteriormente. Por ejemplo, el módulo 870 de codificación implementa, procesa, prepara o proporciona las diversas funciones de red. Por lo tanto, la inclusión del módulo 870 de codificación proporciona una mejora sustancial de la funcionalidad del dispositivo 800 de codificación de vídeo y efectúa una transformación del dispositivo 800 de codificación de vídeo a un estado diferente. Alternativamente, el módulo 870 de codificación se implementa como instrucciones almacenadas en la memoria 860 y ejecutadas por el procesador 830.

El dispositivo 800 de codificación de vídeo también puede incluir dispositivos de entrada y/o salida (E/S) 880 para comunicar datos a y desde un usuario. Los dispositivos de E/S 880 pueden incluir dispositivos de salida, como una pantalla para mostrar datos de vídeo, altavoces para emitir datos de audio, etc. Los dispositivos de E/S 880 también pueden incluir dispositivos de entrada, 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 puede utilizarse como dispositivo de almacenamiento de datos de desbordamiento, para almacenar programas cuando dichos programas se seleccionan para su ejecución, y para almacenar instrucciones y datos que se leen durante la ejecución del programa. La memoria 860 puede ser volátil y/o no volátil y puede ser memoria de sólo lectura (ROM), memoria de acceso aleatorio (RAM), memoria ternaria direccionable por contenido (TCAM), y/o memoria estática de acceso aleatorio (SRAM).

- FIG. 9 es un diagrama esquemático de una realización de un medio 900 de codificación. En la realización, los medios 900 de codificación se implementan en un dispositivo 902 de codificación de vídeo (por ejemplo, un codificador 20 de vídeo o un decodificador 30 de vídeo). El dispositivo 902 de codificación de vídeo incluye medios 901 de recepción. Los medios 901 de recepción están configurados para recibir una imagen para codificar o para recibir un flujo de bits para decodificar. El dispositivo 902 de codificación de vídeo incluye medios 907 de transmisión acoplados a los medios 901 de recepción. Los medios 907 de transmisión están configurados para transmitir el flujo de bits a un decodificador o para transmitir una imagen decodificada a un medio de visualización (por ejemplo, uno de los dispositivos de E/S 880).
- 5
- 10 El dispositivo 902 de codificación de vídeo incluye un medio 903 de almacenamiento. El medio 903 de almacenamiento está acoplado a al menos uno de los medios 901 de recepción o de los medios 907 de transmisión. El medio 903 de almacenamiento está configurado para almacenar instrucciones. El dispositivo 902 de codificación de vídeo también incluye medios 905 de procesamiento. Los medios 905 de procesamiento están acoplados a los medios 903 de almacenamiento. El medio 905 de procesamiento se configura para ejecutar las instrucciones almacenadas en el medio 903 de almacenamiento para realizar los métodos divulgados en el presente documento.
- 15

También debe entenderse que los pasos de los métodos ejemplares expuestos en el presente documento no necesariamente deben realizarse en el orden descrito, y el orden de los pasos de dichos métodos debe entenderse como meramente ejemplar. Asimismo, pueden incluirse pasos adicionales en dichos métodos, y ciertos pasos pueden omitirse o combinarse, en métodos coherentes con diversas realizaciones de la presente divulgación.

20

Además, las técnicas, sistemas, subsistemas y métodos descritos e ilustrados en las diversas realizaciones como discretos o separados pueden combinarse o integrarse con otros sistemas, módulos, técnicas o métodos sin apartarse del alcance de la presente divulgación. Otros elementos mostrados o discutidos como acoplados o directamente acoplados o comunicados entre sí pueden estar indirectamente acoplados o comunicados a través de alguna interfaz, dispositivo o componente intermedio ya sea eléctrica, mecánica o de otro modo. Otros ejemplos de cambios, sustituciones y alteraciones son determinables por un experto en la técnica y podrían realizarse sin apartarse del alcance divulgado en el presente documento.

25

30

REIVINDICACIONES

1. Un método de decodificación de un flujo de bits de vídeo codificado implementado por un decodificador de vídeo, que comprende:
- 5 analizar, a partir del flujo de bits de vídeo codificado, una bandera;
- analizar, a partir del flujo de bits de vídeo codificado, una `ref_pic_list_sps_flag[0]` y una `ref_pic_list_idx[0]` en una primera estructura de sintaxis de lista de imágenes de referencia;
- 10 determinar que una `ref_pic_list_sps_flag[1]` y una `ref_pic_list_idx[1]` para una segunda estructura de sintaxis de lista de imágenes de referencia no están presentes en una cabecera de fragmento del flujo de bits de vídeo codificado y los valores del `ref_pic_list_sps_flag[1]` y `ref_pic_list_idx[1]` de la segunda estructura de sintaxis de lista de imágenes de referencia son los mismos que los valores de `ref_pic_list_sps_flag[0]` y `ref_pic_list_idx[0]` de la primera estructura de sintaxis de lista de imágenes de referencia, respectivamente, cuando la bandera tiene un primer valor;
- 15 donde
- 20 la `ref_pic_list_sps_flag[i]` igual a 1 especifica que una lista de imágenes de referencia `i` de una imagen actual se deriva con base en una de las estructuras de sintaxis de lista de imágenes de referencia con un `listIdx` igual a `i` en un conjunto de parámetros de secuencia activa, SPS; la `ref_pic_list_sps_flag[i]` igual a 0 especifica que la lista de imágenes de referencia `i` de la imagen actual se deriva con base en una estructura de sintaxis de lista de imágenes de referencia con el `listIdx` igual a `i` que se incluye directamente en la cabecera de fragmento;
- 25 `ref_pic_list_idx[i]` especifica un índice, en una lista de estructuras de sintaxis de lista de imágenes de referencia con `listIdx` igual a `i` incluidas en el SPS activo, de la estructura de sintaxis de lista de imágenes de referencia con `listIdx` igual a `i` que se utiliza para la derivación de la lista de imágenes de referencia `i` de la imagen actual;
- 30 `i` igual a 0 o 1;
- analizar, desde la cabecera de fragmento, la `ref_pic_list_sps_flag[1]` y la `ref_pic_list_idx[1]` a la segunda estructura de sintaxis de lista de imágenes de referencia cuando la bandera tiene un segundo valor;
- 35 generar una lista de imágenes de referencia utilizando al menos una de las estructuras sintácticas de la primera lista de imágenes de referencia o de la segunda lista de imágenes de referencia a partir del flujo de bits de vídeo codificado; y
- 40 realizar una interpredicción con base en la lista de imágenes de referencia para generar un bloque reconstruido.
2. El método de la reivindicación 1, en el que la bandera se designa `rpl1_idx_present_flag`.
3. El método de cualquiera de las reivindicaciones 1 a 2, en el que el indicador se incluye en un conjunto de parámetros de imagen, PPS, del flujo de bits de vídeo codificado.
- 45 4. El método de cualquiera de las reivindicaciones 1 a 3, en el que la segunda estructura de sintaxis de lista de imágenes de referencia se incluye en la cabecera de fragmento del flujo de bits de vídeo codificado.
5. El método de la reivindicación 1, en el que la bandera se incluye en un conjunto de parámetros de imagen, PPS, del flujo de bits de vídeo codificado y la segunda estructura de sintaxis de lista de imágenes de referencia se incluye en la cabecera de fragmento del flujo de bits de vídeo codificado.
- 50 6. El método de cualquiera de las reivindicaciones 1 a 5, en el que la `ref_pic_list_sps_flag[1]` y la `ref_pic_list_idx[1]` no se incluyen en la cabecera de fragmento cuando el primer valor de la bandera es cero .
- 55 7. El método de la reivindicación 6, en el que la `ref_pic_list_sps_flag[0]` y la `ref_pic_list_idx[0]` se incluyen en la cabecera de fragmento cuando el segundo valor de la bandera es uno.
8. Método de codificación de un flujo de bits de vídeo implementado por un codificador de vídeo, que comprende:
- 60 determinar si una `ref_pic_list_sps_flag[1]` y una `ref_pic_list_idx[1]` a una segunda estructura de sintaxis de lista de imágenes de referencia está presente en una cabecera de fragmento en el flujo de bits codificado;
- 65 codificar una bandera con un primer valor en el flujo de bits de vídeo cuando se determina que la

5 ref_pic_list_sps_flag[1] y la ref_pic_list_idx[1] de la segunda estructura de sintaxis de lista de imágenes de referencia no están presentes en la cabecera de fragmento del flujo de bits de vídeo y los valores de ref_pic_list_sps_flag[1] y ref_pic_list_idx[1] de la segunda estructura de sintaxis de lista de imágenes de referencia son los mismos que los valores de ref_pic_list_sps_flag[0] y ref_pic_list_idx[0] de una primera estructura de sintaxis de lista de imágenes de referencia, respectivamente;

donde

10 la ref_pic_list_sps_flag[i] igual a 1 especifica que una lista de imágenes de referencia i de una imagen actual se deriva con base en una de las estructuras de sintaxis de lista de imágenes de referencia con un listIdx igual a i en un conjunto de parámetros de secuencia activa, SPS; el indicador ref_pic_list_sps_flag[i] igual a 0 especifica que la lista de imágenes de referencia i de la imagen actual se deriva con base en una estructura de sintaxis de lista de imágenes de referencia con un listIdx igual a i que se incluye directamente en la cabecera de fragmento;

15 ref_pic_list_idx[i] especifica un índice, en una lista de estructuras de sintaxis de lista de imágenes de referencia con listIdx igual a i incluidas en el SPS activo, de la estructura de sintaxis de lista de imágenes de referencia con listIdx igual a i que se utiliza para la derivación de la lista de imágenes de referencia i de la imagen actual;

20 i igual a 0 o 1; codificar la bandera con un segundo valor en el flujo de bits de vídeo al determinar que la ref_pic_list_sps_flag[1] y la ref_pic_list_idx[1] a la segunda estructura de sintaxis de lista de imágenes de referencia están presentes en la cabecera de fragmento del flujo de bits de vídeo;

25 codificar la ref_pic_list_sps_flag[0] y la ref_pic_list_idx[0] a la primera estructura de sintaxis de lista de imágenes de referencia en el flujo de bits de vídeo cuando la bandera se codifica con el primer valor;

30 codificar la ref_pic_list_sps_flag[0] y la ref_pic_list_idx[0] en la primera estructura de sintaxis de lista de imágenes de referencia y la ref_pic_list_sps_flag[1] y la ref_pic_list_idx[1] en la segunda estructura de sintaxis de lista de imágenes de referencia en el flujo de bits de vídeo cuando la bandera se codifica con el segundo valor.

9. El método de la reivindicación 8, en el que el método comprende además transmitir el flujo de bits de vídeo hacia un decodificador de vídeo.

35 10. El método de cualquiera de las reivindicaciones 8 a 9, en el que la bandera está codificada en un conjunto de parámetros de imagen (PPS) del flujo de bits de vídeo codificado.

40 11. Un medio de almacenamiento no transitorio que incluye un flujo de bits codificado, el flujo de bits comprende una pluralidad de estructuras de sintaxis de lista de imágenes de referencia, una ref_pic_list_sps_flag[0] y una ref_pic_list_idx[0] para una primera estructura de sintaxis de lista de imágenes de referencia, y una bandera para especificar si una ref_pic_list_sps_flag[1] y una ref_pic_list_idx[1] para una segunda estructura de sintaxis de lista de imágenes de referencia están presentes en una cabecera de fragmento en el flujo de bits codificado; en el que ref_pic_list_sps_flag[i] igual a 1 especifica que una lista de imágenes de referencia i de una imagen actual se deriva con base en una de las estructuras de sintaxis de lista de imágenes de referencia con un listIdx igual a i en un conjunto de parámetros de secuencia activa, SPS; ref_pic_list_sps_flag[i] igual a 0 especifica que la lista de imágenes de referencia i de la imagen actual se deriva con base en una estructura de sintaxis de lista de imágenes de referencia con el listIdx igual a i que está incluida directamente en la cabecera de fragmento; ref_pic_list_idx[i] especifica un índice, en una lista de estructuras de sintaxis de lista de imágenes de referencia con el listIdx igual a i incluido en el SPS activo, de la estructura de sintaxis de lista de imágenes de referencia con el listIdx igual a i que se utiliza para la derivación de la lista de imágenes de referencia i de la imagen actual; i igual a 0 o 1.

55 12. Un codificador que comprende circuitos de procesamiento para llevar a cabo el método de acuerdo con cualquiera de las reivindicaciones 8 a 10.

13. Un decodificador que comprende circuitos de procesamiento para llevar a cabo el método de acuerdo con cualquiera de las reivindicaciones 1 a 7.

60 14. Un producto de programa de ordenador que comprende un código de programa para realizar el método de acuerdo con cualquiera de las reivindicaciones 1 a 10 cuando se ejecuta en un ordenador o un procesador.

15. Producto de la reivindicación 14 incorporado en un medio de almacenamiento no transitorio legible por ordenador.

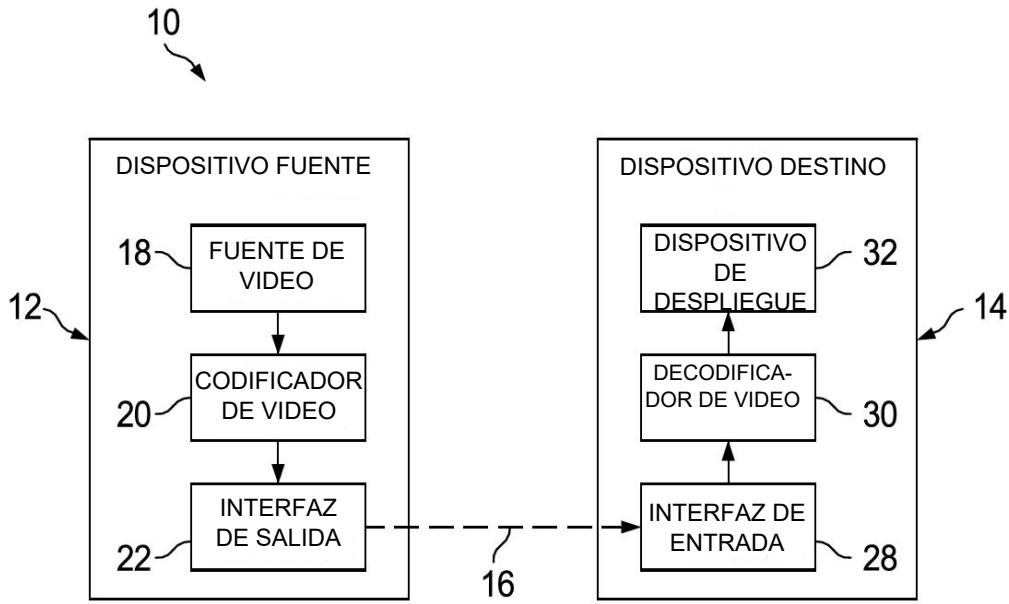


FIGURA 1

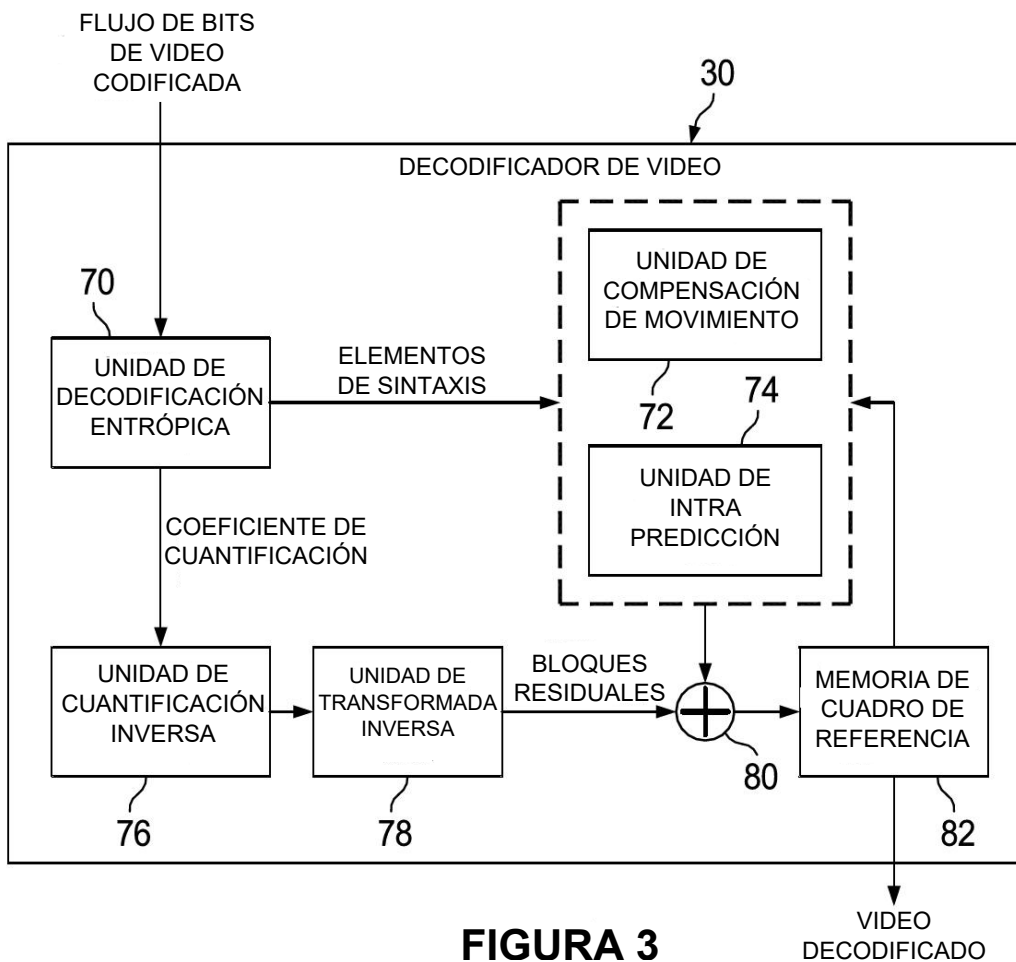


FIGURA 3

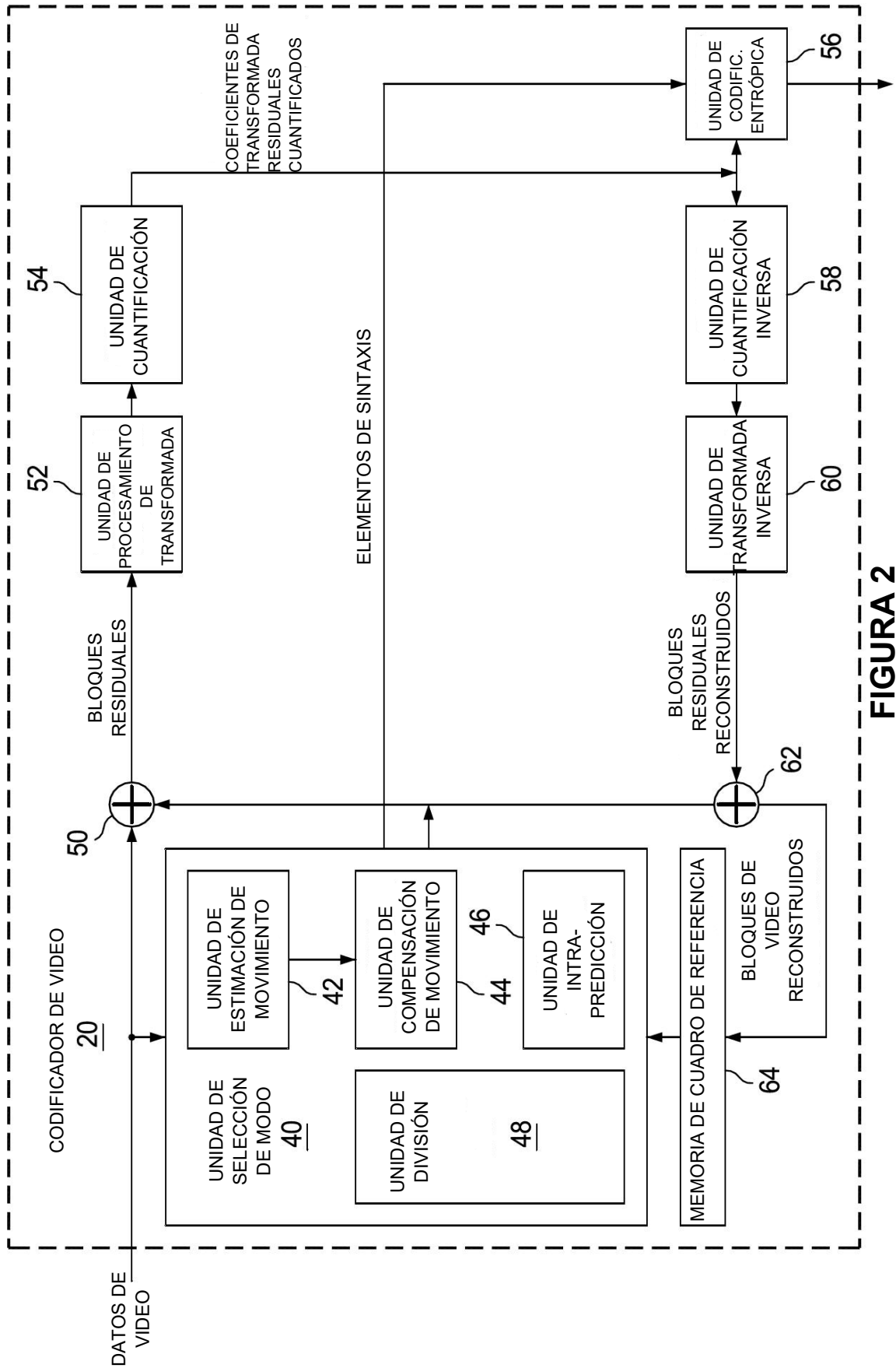


FIGURA 2

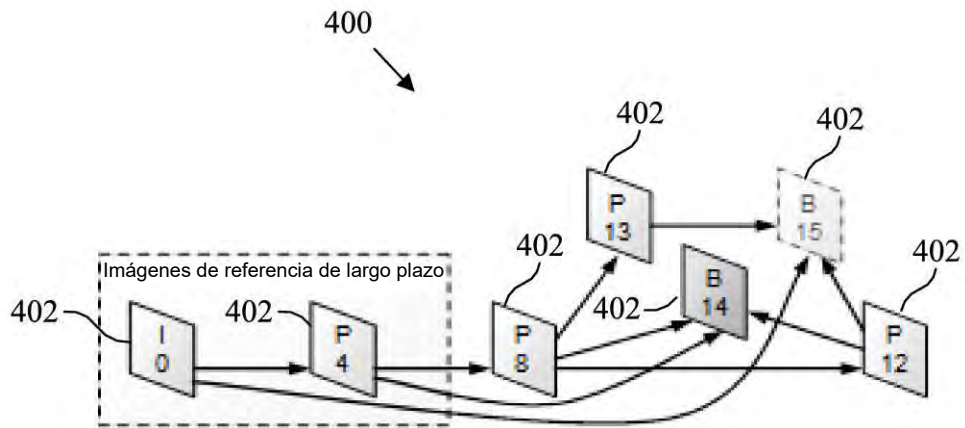


FIGURA 4

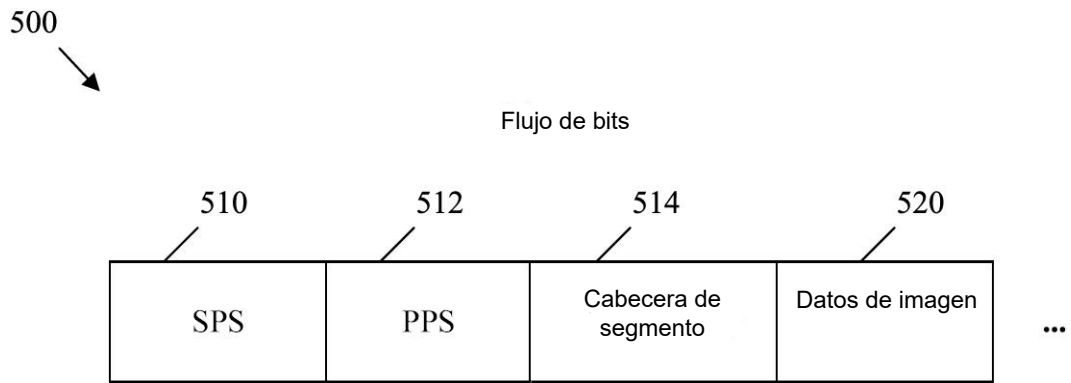


FIGURA 5

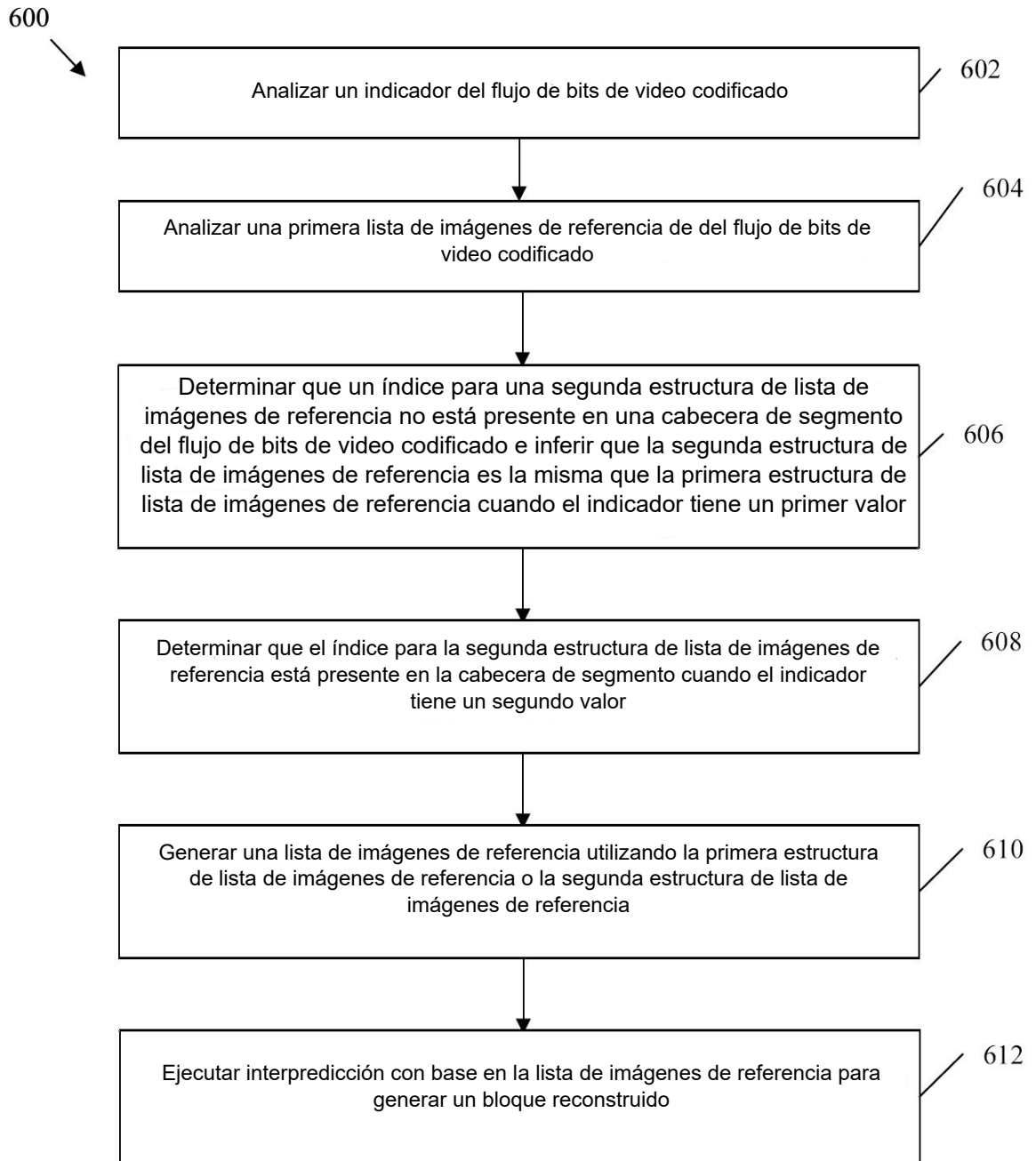


FIGURA 6

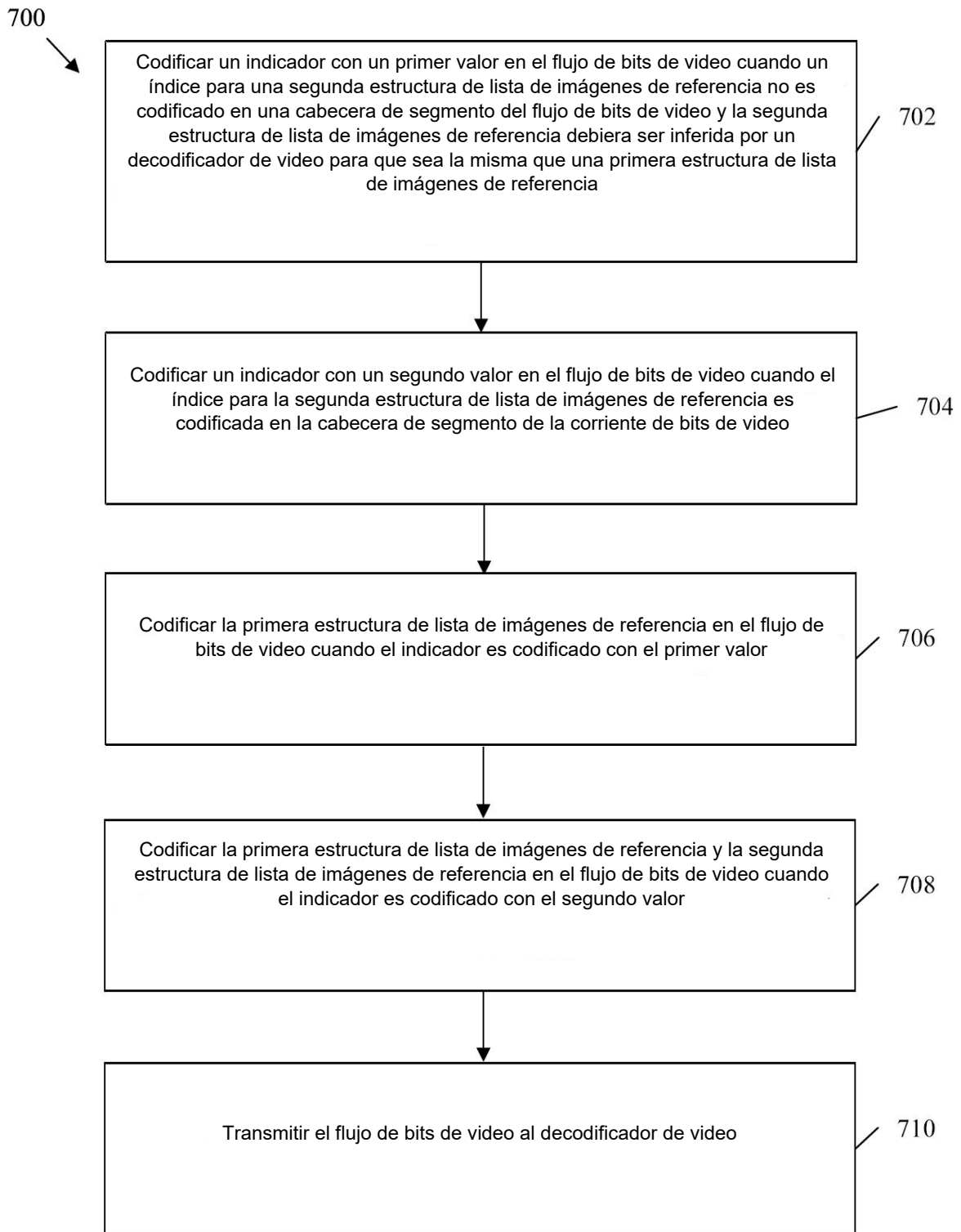


FIGURA 7

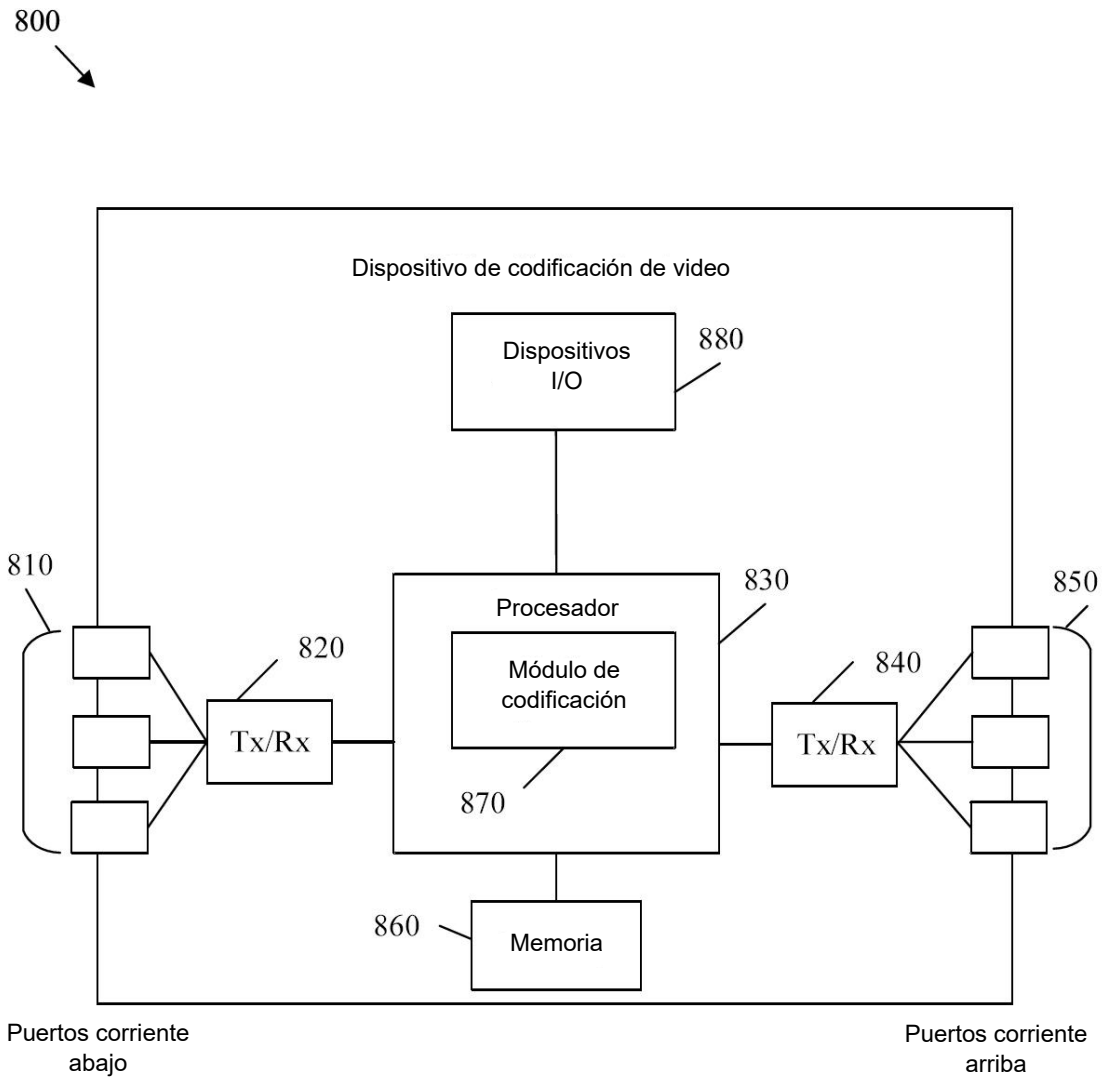


FIGURA 8

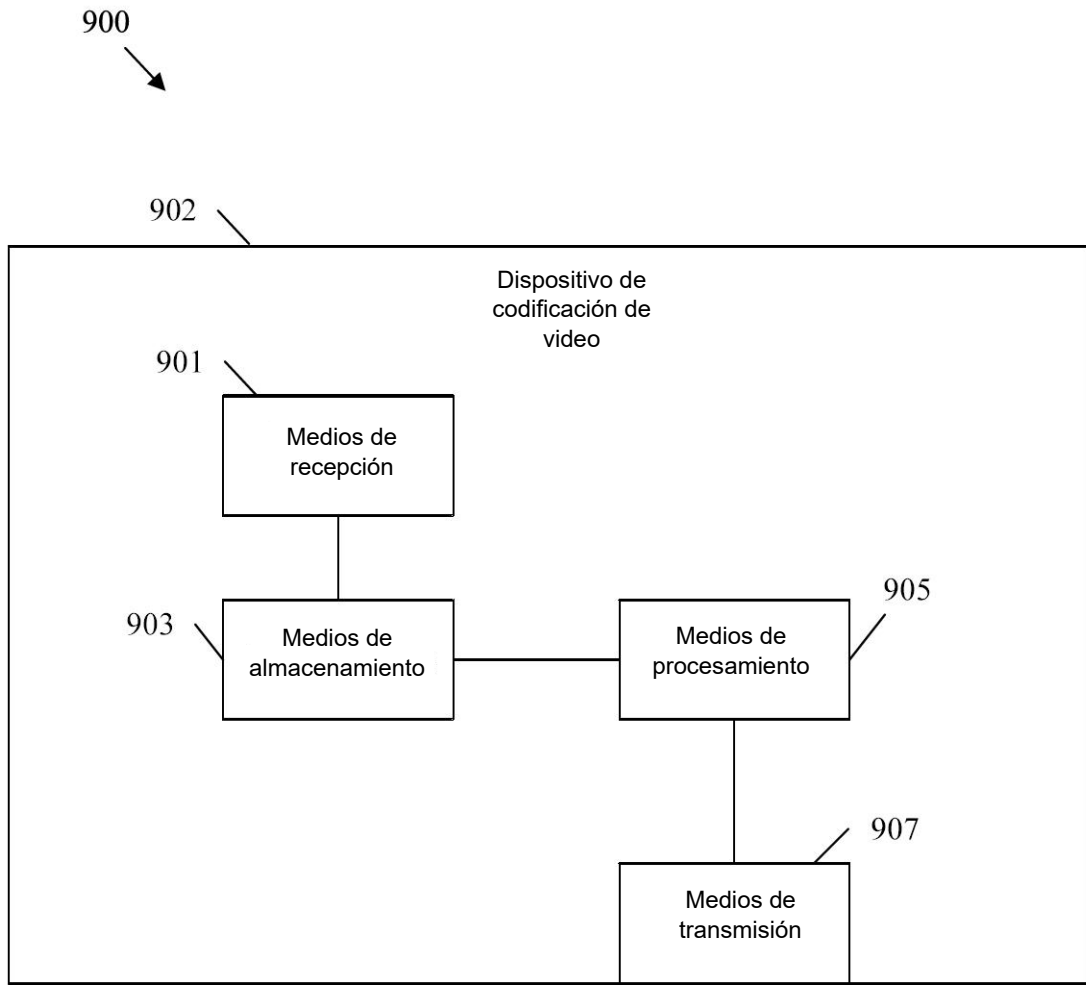


FIGURA 9