

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 633 651**

51 Int. Cl.:

H04N 19/46 (2006.01)

H04N 19/463 (2006.01)

H04N 19/70 (2006.01)

H04N 19/30 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **11.01.2013 PCT/US2013/021227**

87 Fecha y número de publicación internacional: **18.07.2013 WO13106705**

96 Fecha de presentación y número de la solicitud europea: **11.01.2013 E 13700835 (5)**

97 Fecha y número de publicación de la concesión europea: **19.04.2017 EP 2803193**

54 Título: **Codificación de conjuntos de parámetros y cabeceras de unidad NAL para codificación de vídeo**

30 Prioridad:

14.01.2012 US 201261586777 P

16.01.2012 US 201261587070 P

19.01.2012 US 201261588629 P

23.04.2012 US 201261637195 P

24.04.2012 US 201261637774 P

10.01.2013 US 201313738377

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
22.09.2017

73 Titular/es:

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

72 Inventor/es:

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

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 633 651 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

Codificación de conjuntos de parámetros y cabeceras de unidad NAL para codificación de vídeo

5 CAMPO TÉCNICO

Esta divulgación se refiere a la codificación de vídeo.

ANTECEDENTES

- 10 Las capacidades del vídeo digital pueden incorporarse en una amplia gama de dispositivos, incluyendo televisores digitales, sistemas de difusión digital directa, sistemas de difusión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, ordenadores de tableta, lectores de libros electrónicos, cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos celulares o de radio por satélite, los denominados "teléfonos inteligentes", dispositivos de videoconferencia, dispositivos de transmisión de vídeo y similares. Los dispositivos de vídeo digital implementan técnicas de codificación de vídeo, tales como las descritas en las normas definidas por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Parte 10, Codificación Avanzada de Vídeo (AVC), la norma de Codificación de Vídeo de Alta Eficiencia (HEVC), actualmente en desarrollo y las ampliaciones de tales normas. Un reciente borrador de la
- 15 próxima norma HEVC está disponible en http://phenix.int-evry.fr/jct/doc_end_user/documents/7_Geneva/wg11/JCTVC-G1103-v3.zip. Los dispositivos de vídeo pueden transmitir, recibir, codificar, decodificar y/o almacenar información de vídeo digital más eficazmente, implementando tales técnicas de codificación de vídeo.
- 20 Las técnicas de codificación de vídeo incluyen la predicción espacial (intra-imagen) y/o la predicción temporal (entre imágenes) para reducir o eliminar la redundancia intrínseca en las secuencias de vídeo. Para la codificación de vídeo basada en bloques, un fragmento de vídeo (por ejemplo, una trama de vídeo o una parte de una trama de vídeo) puede dividirse en bloques de vídeo, que también pueden denominarse bloques arbolados, unidades de codificación (CU) y/o nodos de codificación. Los bloques de vídeo en un fragmento intracodificado (I) de una imagen son codificados usando la predicción espacial con respecto a muestras de referencia en bloques contiguos de la misma imagen. Los bloques de vídeo en un fragmento intercodificado (P o B) de una imagen pueden usar la predicción espacial con respecto a muestras de referencia en bloques contiguos de la misma imagen, o la predicción temporal con respecto a muestras de referencia en otras imágenes de referencia. Las imágenes pueden denominarse tramas y las imágenes de referencia pueden denominarse tramas de referencia.
- 25 La predicción espacial o temporal da como resultado un bloque predictivo para un bloque a codificar. Los datos residuales representan diferencias de píxeles entre el bloque original a codificar y el bloque predictivo. Un bloque intercodificado se codifica de acuerdo con un vector de movimiento que apunta a un bloque de muestras de referencia que forman el bloque predictivo, y los datos residuales que indican la diferencia entre el bloque codificado y el bloque predictivo. Un bloque intracodificado se codifica de acuerdo con una modalidad de intracodificación y los datos residuales. Para una mayor compresión, los datos residuales pueden transformarse desde el dominio de píxeles a un dominio de transformación, dando como resultado coeficientes de transformación residuales, los cuales pueden cuantizarse posteriormente. Los coeficientes de transformación cuantizados, inicialmente dispuestos en una formación bidimensional, pueden explorarse con el fin de producir un vector unidimensional de coeficientes de transformación, y puede aplicarse la codificación por entropía para lograr aún más compresión. El documento US 2005/0254575 A1 describe la producción de una o más capas del flujo de datos escalable. Las capas se caracterizan por una propiedad de codificación y el procedimiento incluye señalar las capas con la propiedad de codificación de tal manera que sean legibles por un decodificador para determinar la propiedad de codificación sin analizar el flujo de datos escalable. También describe un flujo de bits escalable, en el que al menos dos capas de escalabilidad están presentes, y cada capa está caracterizada por un conjunto de al menos una propiedad, como perfil, nivel y un conjunto de al menos un parámetro HRD / VBV, que puede ser diferente al de la totalidad del flujo, y en el que dicho conjunto de al menos una propiedad está señalado para al menos una capa que es diferente al flujo completo. El documento de Jill Boyce y col., "Extensible High Layer Syntax for Scalability" (Sintaxis de Capa Alta Extensible para Escalabilidad), JCTVC-E279, del 22 de marzo de 2011, divulga emplear un conjunto de parámetros de dependencia DPS además del conjunto de parámetros de secuencia SPS. En el DPS se transmiten los mismos parámetros para múltiples capas y en el SPS se transmiten parámetros que son válidos solo para una capa específica. La invención está definida en la reivindicación adjunta, a la cual se debería hacer referencia en este punto.

60 SUMARIO

- En general, esta divulgación describe técnicas para codificar conjuntos de parámetros y unidades de capa de abstracción de red (NAL) para codificación de vídeo. Estas técnicas pueden aplicarse a datos codificados de capa única, tales como datos de vídeo bidimensionales, así como a datos de vídeo de codificación de vídeo escalable (SVC) y datos de vídeo de codificación de vídeo multivista (MVC). De este modo, los conjuntos de parámetros y las unidades NAL pueden ser mutuamente compatibles entre diversos tipos de datos de vídeo. Por ejemplo, un

codificador de vídeo, tal como un codificador de vídeo o descodificador de vídeo, puede codificar un conjunto de parámetros de vídeo (VPS) que define parámetros para una pluralidad de capas de datos de vídeo. Las capas pueden corresponder, por ejemplo, a capas SVC (que tienen varias velocidades de trama, resoluciones espaciales y/o niveles de calidad) y/o vistas de datos MVC (por ejemplo, secuencias de imágenes de una escena capturada desde varias perspectivas de cámara sobre un eje horizontal).

En un ejemplo, un procedimiento de codificación de datos de vídeo incluye codificar un conjunto de parámetros de vídeo (VPS) para una pluralidad de capas de datos de vídeo, en el que cada una de la pluralidad de capas de datos de vídeo se refiere al VPS y codificar una pluralidad de capas de datos de vídeo basadas al menos en parte en el VPS, y en el que la codificación del VPS comprende:
codificación de datos del VPS indicativos de un número máximo de capas temporales en la pluralidad de capas de datos de vídeo.

En otros ejemplos, se proporciona un dispositivo correspondiente al procedimiento anterior y un medio de almacenamiento legible por ordenador que ha almacenado en él instrucciones que, cuando se ejecutan, realizan el procedimiento anterior.

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

BREVE DESCRIPCIÓN DE LOS DIBUJOS

La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación y descodificación de vídeo de ejemplo que puede utilizar técnicas para codificar conjuntos de parámetros y unidades de capa de abstracción de red (NAL) para una o más capas de datos de vídeo

La FIG. 2 es un diagrama de bloques que ilustra un ejemplo del codificador de vídeo 20 que puede implementar las técnicas de codificación de conjuntos de parámetros y unidades NAL para una o más capas de datos de vídeo.

La FIG. 3 es un diagrama de bloques que ilustra un ejemplo del descodificador de vídeo 30 que puede implementar técnicas para codificación de conjuntos de parámetros y unidades NAL para una o más capas de datos de vídeo.

La FIG. 4 es un diagrama conceptual que ilustra un patrón de predicción de MVC a modo de ejemplo.

La FIG. 5 es un diagrama conceptual que ilustra un conjunto de parámetros de vídeo (VPS) y varios conjuntos de parámetros de capa (LPS).

La FIG. 6 es un diagrama conceptual que ilustra un ejemplo de un conjunto de parámetros de agrupación (GPS) y relaciones del GPS con otros conjuntos de parámetros y cabeceras de fragmentos.

La FIG. 7 es un diagrama de flujo que ilustra un ejemplo de procedimiento de codificación de datos de vídeo de acuerdo con las técnicas de esta divulgación.

La FIG. 8 es un diagrama de flujo que ilustra un ejemplo de procedimiento de descodificación de datos de vídeo de acuerdo con las técnicas de esta divulgación.

La FIG. 9 es un diagrama de flujo que ilustra un ejemplo de procedimiento de codificación de datos de vídeo basándose, al menos en parte, en un cierto número de capas temporales como se señala en un VPS.

La FIG. 10 es un diagrama de flujo que ilustra un ejemplo de procedimiento de codificación de datos de vídeo basándose, al menos en parte, en un cierto número de imágenes a reordenar en una o más capas e imágenes a almacenar en una memoria intermedia de imágenes descodificadas.

La FIG. 11 es un diagrama de flujo que ilustra un ejemplo de procedimiento de codificación de datos de vídeo basándose, al menos en parte, en parámetros de descodificador de referencia hipotéticos (HRD) señalados en un VPS.

La FIG. 12 es un diagrama de flujo que ilustra un ejemplo de procedimiento de codificación de datos de vídeo basándose, al menos en parte, en datos de ampliación señalados en un VPS.

DESCRIPCIÓN DETALLADA

En general, esta divulgación describe la codificación de datos de vídeo usando un conjunto de parámetros de vídeo

(VPS). Los datos de vídeo pueden clasificarse jerárquicamente como incluyendo una pluralidad de capas, una secuencia de imágenes dentro de una capa dada, una imagen dentro de una secuencia, fragmentos dentro de una imagen y bloques (por ejemplo, macrobloques o unidades de árbol de codificación) dentro de un fragmento. Pueden usarse conjuntos de parámetros de secuencia (SPS) para señalar los parámetros cambiantes con poca frecuencia para una secuencia de imágenes, y se pueden usar conjuntos de parámetros de imagen (PPS) para señalar los parámetros cambiantes con poca frecuencia de las imágenes individuales.

De acuerdo con las técnicas de esta divulgación, un VPS puede señalar los parámetros cambiantes con poca frecuencia para una pluralidad de secuencias a través de las capas respectivas. Es decir, un VPS puede incluir parámetros para un conjunto de secuencias temporalmente coubicadas de capas diferentes. Diferentes capas pueden incluir, por ejemplo, vistas diferentes para datos de vídeo multivista, capas de diferentes calidades, capas de resolución espacial diferentes, capas escalables temporalmente (es decir, capas que permiten diferentes velocidades de trama) y similares. De esta manera, se puede proporcionar un VPS para una pluralidad de capas diferentes, de manera que el VPS señale los parámetros que son comunes a cada una de las capas respectivas (por ejemplo, secuencias respectivas dentro de las capas respectivas). Se puede decir que un flujo de bits incluye cada una de la pluralidad de capas, y las capas respectivas pueden formar flujos secundarios de bits respectivos. Además, un flujo secundario de bits puede corresponder a una combinación de dos o más capas.

Esta divulgación describe varios ejemplos de datos que pueden incluirse en un VPS. Dichos datos pueden incluir, en algunos ejemplos, una indicación de un cierto número de subcapas (por ejemplo, un número máximo de subcapas) dentro de las capas correspondientes. Por ejemplo, un VPS puede incluir datos que señalan un cierto número de capas temporales y/o un número máximo de capas temporales (por ejemplo, un identificador de capa temporal más alto).

Para mencionar otro ejemplo, un VPS puede incluir, adicionalmente o de forma alternativa, datos sustancialmente similares a cualquier dato previamente señalado en un SPS (es decir, señalado en SPS convencionales). De esta manera, cuando las secuencias de dos o más capas de un flujo de bits incluyen parámetros sustancialmente similares o idénticos, un codificador de vídeo puede codificar un VPS para señalar parámetros para las secuencias de las capas, en lugar de codificar redundantemente dichos datos en SPS respectivos para las diversas secuencias entre las diferentes capas.

Un VPS puede, adicionalmente o de forma alternativa, incluir datos que definen información de usabilidad de vídeo (VUI), tales como información de representación de vídeo, parámetros de descodificador de referencia hipotéticos (HRD) y/o información de restricción del flujo de bits. La información de restricción del flujo de bits puede incluir restricciones en el rango del vector de movimiento, el tamaño de la memoria intermedia de imágenes descodificadas (DPB) (por ejemplo, en términos de un cierto número de imágenes que se mantendrán en la DPB), el número de tramas de reordenación (es decir, una indicación de un cierto número de tramas a ser reordenadas del orden de descodificación al orden de visualización), tamaños codificados de bloques (por ejemplo, macrobloques (MBs) o unidades de árbol de codificación) y tamaños codificados de imágenes. Un VPS puede proporcionar datos adicionales para una o más ampliaciones del VPS, de manera que el VPS pueda ampliarse mediante futuras normas o ampliaciones de la próxima norma HEVC.

La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación y descodificación de vídeo de ejemplo 10 que puede utilizar técnicas para codificar conjuntos de parámetros y unidades de capa de abstracción de red (NAL) para una o más capas de datos de vídeo. Como se muestra en la FIG. 1, el sistema 10 incluye un dispositivo fuente 12 que proporciona datos de vídeo codificados, a descodificar en un momento posterior mediante un dispositivo de destino 14. En particular, el dispositivo fuente 12 proporciona los datos de vídeo al dispositivo de destino 14 a través de un medio legible por ordenador 16. El dispositivo fuente 12 y el dispositivo de destino 14 pueden comprender cualquiera entre una amplia gama de dispositivos, incluyendo ordenadores de sobremesa, ordenadores plegables (es decir, portátiles), ordenadores de tableta, descodificadores, equipos telefónicos de mano tales como los denominados teléfonos "inteligentes", los denominados paneles "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, un dispositivo de transmisión de vídeo o similares. En algunos casos, el dispositivo fuente 12 y el dispositivo de destino 14 pueden estar equipados para la comunicación inalámbrica.

El dispositivo de destino 14 puede recibir los datos de vídeo codificados que se van a descodificar mediante el medio legible por ordenador 16. El medio legible por ordenador 16 puede comprender cualquier tipo de medio o dispositivo capaz de desplazar los datos de vídeo codificados desde el dispositivo fuente 12 al dispositivo de destino 14. En un ejemplo, el medio legible por ordenador 16 puede comprender un medio de comunicación para permitir al dispositivo fuente 12 transmitir datos de vídeo codificados directamente al dispositivo de destino 14 en tiempo real. Los datos de vídeo codificados pueden ser modulados de acuerdo con una norma de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitidos al dispositivo de destino 14. El medio de comunicación puede comprender cualquier medio de comunicación inalámbrico o cableado, tal como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión física. El medio de comunicación puede formar parte de una red basada en paquetes, tal como una red de área local, una red de área extensa o una red global tal 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 fuente 12 al dispositivo de destino 14.

En algunos ejemplos, pueden emitirse datos codificados desde la interfaz de salida 22 hasta un dispositivo de almacenamiento. De forma similar, se puede acceder a los datos codificados del dispositivo de almacenamiento mediante una interfaz de entrada. El dispositivo de almacenamiento puede incluir cualquiera entre una diversidad de medios de almacenamiento de datos, de acceso distribuido o local, tales como una unidad de disco rígido, discos Blu-ray, discos DVD, discos CD-ROM, memoria flash, memoria volátil o no volátil, u otros medios adecuados cualesquiera de almacenamiento digital, para almacenar datos de vídeo codificados. En un ejemplo adicional, el dispositivo de almacenamiento puede corresponder a un servidor de ficheros o a otro dispositivo de almacenamiento intermedio que pueda almacenar los datos de vídeo codificados, generados por el dispositivo fuente 12. El dispositivo de destino 14 puede acceder a datos de vídeo almacenados del dispositivo de almacenamiento, mediante flujo o descarga. El servidor de ficheros puede ser cualquier tipo de servidor capaz de almacenar datos de vídeo codificados y transmitir esos datos de vídeo codificados al dispositivo de destino 14. Los servidores de ficheros a modo de ejemplo incluyen un servidor de Internet (por ejemplo, para una sede en la Red), un servidor del FTP, dispositivos de almacenamiento conectados a la red (NAS) o un controlador de disco local. El dispositivo de destino 14 puede acceder a los datos de vídeo codificado a través de cualquier conexión de datos estándar, incluyendo una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión de Wi-Fi), una conexión por cable (por ejemplo, DSL, módem de cable, etc.), o una combinación de ambos que sea adecuada para acceder a datos de vídeo codificado, almacenados en un servidor de ficheros. La transmisión de datos de vídeo codificados desde el dispositivo de almacenamiento puede ser una transmisión por flujo, una transmisión de descarga o una combinación de ambas.

Las técnicas de esta divulgación no están limitadas necesariamente a aplicaciones o configuraciones inalámbricas. Las técnicas pueden aplicarse a la codificación de vídeo, en soporte de cualquiera entre una diversidad de aplicaciones de multimedia, tales como difusiones de televisión por el aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones de vídeo por flujo de Internet, tales como el flujo adaptativo dinámico sobre HTTP (DASH), vídeo digital que se codifica en un medio de almacenamiento de datos, descodificación de vídeo digital almacenado en un medio de almacenamiento de datos, u otras aplicaciones. En algunos ejemplos, el sistema 10 puede configurarse para dar soporte a la transmisión de vídeo unidireccional o bidireccional, para prestar soporte a aplicaciones tales como la transmisión de vídeo, la reproducción de vídeo, la difusión de vídeo y/o la videotelefonía.

En el ejemplo de la FIG. 1, el dispositivo fuente 12 incluye una fuente de vídeo 18, un codificador de vídeo 20 y una interfaz de salida 22. El dispositivo de destino 14 incluye una interfaz de entrada 28, un decodificador de vídeo 30 y un dispositivo de visualización 32. De acuerdo con esta divulgación, el codificador de vídeo 20 del dispositivo fuente 12 puede estar configurado para aplicar las técnicas de codificación por entropía conjuntos de parámetros y unidades NAL para una o más capas de datos de vídeo. En otros ejemplos, un dispositivo fuente y un dispositivo de destino pueden incluir otros componentes o disposiciones. Por ejemplo, el dispositivo fuente 12 puede recibir datos de vídeo desde una fuente de vídeo externa 18 como una cámara externa. Asimismo, el dispositivo destino 14 puede interactuar con un dispositivo de visualización externo, en lugar de incluir un dispositivo de visualización integrado.

El sistema ilustrado 10 de la figura 1 es simplemente un ejemplo. Las técnicas para codificar conjuntos de parámetros y unidades NAL para una o más capas de datos de vídeo pueden ser realizadas por cualquier dispositivo de codificación y/o decodificación de vídeo digital. Aunque, en general, las técnicas de esta divulgación se llevan a cabo mediante un dispositivo de codificación de vídeo, las técnicas también pueden llevarse a cabo mediante un codificador/decodificador de vídeo, denominado típicamente "CÓDEC". Además, las técnicas de esta divulgación también pueden llevarse a cabo mediante un preprocesador de vídeo. El dispositivo fuente 12 y el dispositivo de destino 14 son simplemente ejemplos de tales dispositivos de codificación, donde el dispositivo fuente 12 genera datos de vídeo codificados para su transmisión al dispositivo de destino 14. En algunos ejemplos, los dispositivos 12, 14 pueden funcionar de manera esencialmente simétrica, de modo que cada uno de los dispositivos 12, 14 incluya componentes de codificación y de decodificación de vídeo. Por lo tanto, el sistema 10 puede dar soporte a una transmisión de vídeo unidireccional o bidireccional entre los dispositivos de vídeo 12, 14, por ejemplo, para la transmisión de vídeo, la reproducción de vídeo, la radiodifusión de vídeo o la videotelefonía.

La fuente de vídeo 18 del dispositivo fuente 12 puede incluir un dispositivo de captura de vídeo, como una cámara de vídeo, un archivo de vídeo que contiene vídeo grabado previamente y/o una interfaz de alimentación de vídeo para recibir vídeo de un proveedor de contenido de vídeo. Como una alternativa adicional, la fuente de vídeo 18 puede generar datos basados en gráficos de ordenador como el vídeo de origen, o una combinación de vídeo en directo, vídeo archivado y vídeo generado por ordenador. En algunos casos, si la fuente de vídeo 18 es una videocámara, el dispositivo fuente 12 y el dispositivo de destino 14 pueden formar los denominados teléfonos con cámara o videoteléfonos. Sin embargo, como se ha mencionado anteriormente, las técnicas descritas en esta divulgación pueden ser aplicables a la codificación de vídeo en general, y pueden aplicarse a aplicaciones inalámbricas y/o cableadas. En cada caso, el vídeo grabado, pregrabado o generado por ordenador puede ser codificado por el codificador de vídeo 20. La información de vídeo codificada puede ser entonces emitida por la interfaz de salida 22 en un medio legible por ordenador 16.

El medio legible por ordenador 16 puede incluir medios transitorios, tales como una emisión inalámbrica o una transmisión de red cableada, o medios de almacenamiento (es decir, medios de almacenamiento no transitorio), tales como un disco duro, una unidad de memoria flash, un disco compacto, un disco de vídeo digital, un disco Blu-ray u otro medio legible por ordenador. En algunos ejemplos, un servidor de red (no se muestra) puede recibir datos de vídeo codificados desde el dispositivo fuente 12 y proporcionar los datos de vídeo codificados al dispositivo de destino 14, por ejemplo, mediante transmisión por red. De manera similar, un dispositivo informático de una instalación de producción de un medio, como una instalación de grabación de discos, puede recibir datos de vídeo codificados desde el dispositivo fuente 12 y producir un disco que contiene los datos de vídeo codificados. Por lo tanto, puede entenderse que el medio legible por ordenador 16 incluye uno o más medios legibles por ordenador de varias formas, en varios ejemplos.

La interfaz de entrada 28 del dispositivo de destino 14 recibe información desde el medio legible por ordenador 16. La información del medio legible por ordenador 16 puede incluir información sintáctica definida por el codificador de vídeo 20, que también es usada por el descodificador de vídeo 30, que incluye elementos sintácticos que describen características y/o el procesamiento de bloques y otras unidades codificadas, por ejemplo, grupos de imágenes (GOP). El dispositivo de visualización 32 muestra los datos de vídeo descodificados a un usuario y puede comprender cualquiera entre 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 diodos orgánicos emisores de luz (OLED) u otro tipo de dispositivo de visualización.

El codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con una norma de compresión de vídeo, tal como la norma de codificación de vídeo de alta eficacia (HEVC), actualmente en fase de elaboración, y pueden ajustarse al modelo de prueba HEVC (HM). De forma alternativa, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con otras normas privadas o industriales, tales como la norma ITU-T H.264, también denominada MPEG-4, Parte 10, codificación de vídeo avanzada (AVC) o ampliaciones de dichas normas. Sin embargo, las técnicas de esta divulgación no están limitadas a ninguna norma de codificación particular. Otros ejemplos de normas de codificación de vídeo incluyen MPEG-2 e ITU-T H.263. Aunque no se muestra en la FIG. 1, en algunos aspectos, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden estar integrado, cada uno de ellos, con un codificador y descodificador de audio, y pueden incluir unidades adecuadas de multiplexado y demultiplexado, u otro hardware y software, para gestionar la codificación, tanto de audio como de vídeo, en un flujo de datos común o en flujos de datos diferentes. Si procede, las unidades de multiplexado y demultiplexado pueden ajustarse al protocolo de multiplexado ITU H.223 o a otros protocolos, tales como el protocolo de datagramas de usuario (UDP).

La norma ITU-T H.264 / MPEG-4 (AVC) fue formulada por el Grupo de Expertos en Codificación de Vídeo de ITU-T (VCEG), junto al Grupo de Expertos en Películas de ISO / IEC (MPEG), como el producto de una asociación colectiva conocida como el Equipo de Vídeo Conjunto (JVT). En algunos aspectos, las técnicas descritas en esta divulgación pueden ser aplicadas a dispositivos que se ajustan en general a la norma H.264. La norma H.264 se describe en la Recomendación ITU-T H.264, Codificación de Vídeo Avanzada, para los servicios audiovisuales genéricos, por el Grupo de Estudio de la ITU-T, con fecha de marzo de 2005, y que se puede denominar en el presente documento norma H.264 o memoria descriptiva H.264, o la norma o memoria descriptiva H.264/AVC. El Equipo de Vídeo Conjunto (JVT) continúa trabajando en ampliaciones para H.264/MPEG-4 AVC.

El codificador de vídeo 20 y el descodificador de vídeo 30 pueden implementarse como cualquiera entre una variedad de circuitos de codificadores adecuados, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados de aplicación específica (ASIC), matrices de puertas programables in situ (FPGA), lógica discreta, software, hardware, firmware o cualquier combinación de estos. Cuando las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en unos medios legibles por ordenador no transitorios adecuados, y ejecutar las instrucciones en hardware mediante uno o más procesadores que realizan las técnicas de esta divulgación. Tanto el codificador de vídeo 20 como el descodificador de vídeo 30 pueden estar incluidos en uno o más codificadores o descodificadores, donde cualquiera de ambos puede estar integrado como parte de un codificador/descodificador (CÓDEC) combinado en un dispositivo respectivo.

El equipo JCT-VC está trabajando en la elaboración de la norma HEVC. Las actividades de normalización de la HEVC se basan en un modelo en evolución de un dispositivo de codificación de vídeo denominado modelo de prueba HEVC (HM). El HM supone varias capacidades adicionales de los dispositivos de codificación de vídeo respecto a dispositivos existentes de acuerdo con, por ejemplo, la ITU-T H.264/AVC. Por ejemplo, mientras que la norma H.264 proporciona nueve modos de codificación de intrapredicción, el HM puede proporcionar hasta treinta y tres modos de codificación de intrapredicción.

En general, el modelo de funcionamiento del HM describe que una trama o imagen de vídeo puede dividirse en una secuencia de bloques arbolados o unidades de codificación de mayor tamaño (LCU), que incluyen muestras tanto de luma como de croma. Los datos sintácticos de un flujo de bits pueden definir un tamaño para la LCU, que es la mayor unidad de codificación en lo que respecta al número de píxeles. Un fragmento incluye un cierto número de

bloques arbolados consecutivos en orden de codificación. Una trama o imagen de vídeo puede dividirse en uno o más fragmentos. Cada bloque arbolado puede separarse en unidades de codificación (CU) de acuerdo con un árbol cuádruple. En general, una estructura de datos de árbol cuádruple incluye un nodo por CU, donde un nodo raíz corresponde al bloque arbolado. Si una CU se divide en cuatro sub-CU, el nodo correspondiente a la CU incluye cuatro nodos hoja, cada uno de los cuales corresponde a una de las sub-CU.

Cada nodo de la estructura de datos de árbol cuádruple puede proporcionar datos sintácticos para la CU correspondiente. Por ejemplo, un nodo en el árbol cuádruple puede incluir un indicador de división, que indica si la CU correspondiente al nodo está dividida o no en varias sub-CU. Los elementos sintácticos de una CU pueden definirse de manera recursiva y pueden depender de si la CU está dividida o no en varias sub-CU. Si una CU no está dividida adicionalmente, se denomina CU hoja. En esta divulgación, cuatro sub-CU de una CU hoja también se denominarán CU hoja aunque no haya una división explícita de la CU hoja original. Por ejemplo, si una CU con un tamaño de 16x16 no se divide adicionalmente, las cuatro sub-CU de tamaño 8x8 también se denominarán CU hoja aunque la CU de tamaño 16x16 no se haya dividido nunca.

Una CU tiene un propósito similar a un macrobloque de la norma H.264, excepto que una CU no tiene una distinción de tamaño. Por ejemplo, un bloque arbolado puede dividirse en cuatro nodos secundarios (también denominados sub-CU) y cada nodo secundario puede a su vez ser un nodo principal y dividirse en otros cuatro nodos secundarios. Un nodo secundario final, no dividido, denominado un nodo hoja del árbol cuádruple, comprende un nodo de codificación, también denominado CU hoja. Los datos sintácticos asociados a un flujo de bits codificado pueden definir un número máximo de veces que puede dividirse un bloque arbolado, denominado profundidad de CU máxima, y también pueden definir un tamaño mínimo de los nodos de codificación. Por consiguiente, un flujo de bits también puede definir una mínima unidad de codificación (SCU). Esta divulgación utiliza el término "bloque" para referirse a cualquiera entre una CU, PU o TU, en el contexto de HEVC, o a estructuras de datos similares en el contexto de otras normas (por ejemplo, macrobloques y subbloques de los mismos en H.264/AVC).

Una CU incluye un nodo de codificación y unidades de predicción (PU) y unidades de transformada (TU) asociadas al nodo de codificación. Un tamaño de la CU corresponde a un tamaño del nodo de codificación y debe ser de forma cuadrada. El tamaño de la CU puede variar desde 8 x 8 píxeles hasta el tamaño del bloque arbolado, con un máximo de 64 x 64 píxeles o más. Cada CU puede contener una o más PU y una o más TU. Los datos sintácticos asociados a una CU pueden describir, por ejemplo, la división de la CU en una o más PU. Los modos de división pueden diferir dependiendo de si la CU está codificada en modo de salto o directo, codificada en modo de intrapredicción o codificada en modo de interpredicción. Las PU pueden dividirse para no tener forma cuadrada. Los datos sintácticos asociados a una CU también pueden describir, por ejemplo, la división de la CU en una o más TU de acuerdo con un árbol cuádruple. Una TU puede tener forma cuadrada o no cuadrada (por ejemplo, rectangular).

La norma HEVC admite transformaciones de acuerdo con las TU, que pueden ser diferentes para diferentes CU. El tamaño de las TU típicamente se basa en el tamaño de las PU de una CU dada definida para una LCU dividida, aunque puede que no siempre sea así. Las TU presentan típicamente el mismo tamaño o un tamaño más pequeño que las PU. En algunos ejemplos, las muestras residuales correspondientes a una CU pueden subdividirse en unidades más pequeñas mediante una estructura de árbol cuádruple conocida como "árbol cuádruple residual" (RQT). Los nodos hoja del RQT pueden denominarse unidades de transformada (TU). Los valores de diferencias de píxeles asociados a las TU pueden transformarse para generar coeficientes de transformada, que pueden cuantizarse.

Una CU hoja puede incluir una o más unidades de predicción (PU). En general, una PU representa una zona espacial correspondiente a la totalidad, o una parte de la CU correspondiente, y puede incluir datos para recuperar una muestra de referencia para la PU. Además, una PU incluye datos relacionados con la predicción. Por ejemplo, cuando la PU está codificada en intramodo, pueden incluirse datos para la PU en un árbol cuádruple residual (RQT), que pueden incluir datos que describen un modo de intrapredicción para una TU correspondiente a la PU. Para mencionar otro ejemplo, cuando la PU está codificada en intermodo, la PU puede incluir datos que definen uno o más vectores de movimiento para la PU. Los datos que definen el vector de movimiento para una PU pueden describir, por ejemplo, un componente horizontal del vector de movimiento, un componente vertical del vector de movimiento, una resolución para el vector de movimiento (por ejemplo, precisión de píxeles de un cuarto o precisión de píxeles de un octavo), una imagen de referencia a la que apunta el vector de movimiento y/o una lista de imágenes de referencia (por ejemplo, la Lista 0, la Lista 1 o la Lista C) para el vector de movimiento.

Una CU hoja que tiene una o más PU también puede incluir una o más unidades de transformación (TU). Las unidades de transformación pueden especificarse usando una RQT (también denominada estructura de árbol cuádruple de TU), como la descrita anteriormente. Por ejemplo, un indicador de división puede indicar si una CU hoja está dividida en cuatro unidades de transformación. A continuación, cada unidad de transformación puede dividirse adicionalmente en más sub-TU. Cuando una TU no está dividida adicionalmente, puede denominarse como TU hoja. En general, en lo que respecta a la intracodificación, todas las TU hoja que pertenecen a una CU hoja comparten el mismo modo de intrapredicción. Es decir, el mismo modo de intrapredicción se aplica en general para calcular valores predichos para todas las TU de una CU hoja. En lo que respecta a la intracodificación, un codificador de vídeo puede calcular un valor residual para cada TU hoja usando el modo de intrapredicción, como

una diferencia entre la parte de la CU correspondiente a la TU y el bloque original. Una TU no está necesariamente limitada al tamaño de una PU. De este modo, las TU pueden ser mayores o menores que una PU. En lo que respecta a la intracodificación, una PU puede estar coubicada con una TU hoja correspondiente para la misma CU. En algunos ejemplos, el tamaño máximo de una TU hoja puede corresponder con el tamaño de la CU hoja correspondiente.

Además, las TU de las CU hoja también pueden asociarse a las estructuras de datos respectivas de árbol cuaternario, denominadas árboles cuaternarios residuales (RQTs). Es decir, una CU hoja puede incluir un árbol cuádruple que indica cómo la CU hoja está dividida en varias TU. El nodo raíz de un árbol cuádruple de TU corresponde en general a una CU hoja, mientras que el nodo raíz de un árbol cuádruple de CU corresponde en general a un bloque arbolado (o LCU). Las TU del RQT que no están divididas se denominan TU hoja. En general, esta divulgación usa los términos CU y TU para hacer referencia a una CU hoja y a una TU hoja, respectivamente, a no ser que se indique lo contrario.

Una secuencia de vídeo incluye típicamente una serie de tramas o imágenes de vídeo. Un grupo de imágenes (GOP) comprende en general una serie de una o más de las imágenes de vídeo. Un GOP puede incluir datos sintácticos en una cabecera del GOP, en una cabecera de una o más de las imágenes o en otras ubicaciones, que describen un cierto número de imágenes incluidas en el GOP. Cada fragmento de una imagen puede incluir datos sintácticos de fragmento que describen un modo de codificación para el fragmento respectivo. Un codificador de vídeo 20 actúa típicamente sobre bloques de vídeo de fragmentos de vídeo individuales con el fin de codificar los datos de vídeo. Un bloque de vídeo puede corresponder a un nodo de codificación de una CU. Los bloques de vídeo pueden presentar tamaños fijos o variables y pueden diferir en tamaño de acuerdo con una norma de codificación especificada.

En un ejemplo, el HM admite la predicción en diversos tamaños de PU. Suponiendo que el tamaño de una CU particular sea $2N \times 2N$, el HM admite la intrapredicción en tamaños de PU de $2N \times 2N$ o $N \times N$ y la interpredicción en tamaños de PU simétricos de $2N \times 2N$, $2N \times N$, $N \times 2N$ o $N \times N$. El HM también admite la división asimétrica para la interpredicción en tamaños de PU de $2N \times nU$, $2N \times nD$, $nL \times 2N$ y $nR \times 2N$. En la división asimétrica, una dirección de una CU no está dividida, mientras que la otra dirección está dividida en 25 % y 75 %. La parte de la CU correspondiente a la división de 25 % está indicada por una "n" seguida de una indicación "arriba", "abajo", "izquierda" o "derecha". Así, por ejemplo, " $2N \times nU$ " se refiere a una CU $2N \times 2N$ que está dividida horizontalmente con una PU $2N \times 0,5N$ encima y una PU $2N \times 1,5N$ debajo.

En esta divulgación, " $N \times N$ " y "N por N" pueden usarse indistintamente para hacer referencia a las dimensiones de píxeles de un bloque de vídeo en términos de dimensiones verticales y horizontales, por ejemplo, 16×16 píxeles o 16 por 16 píxeles. En general, un bloque de tamaño 16×16 tendrá 16 píxeles en la dirección vertical ($y = 16$) y 16 píxeles en la dirección horizontal ($x = 16$). Asimismo, un bloque de tamaño $N \times N$ presenta en general N píxeles en una dirección vertical y N píxeles en una dirección horizontal, donde N representa un valor entero no negativo. Los píxeles en un bloque pueden estar dispuestos en filas y columnas. Además, los bloques no necesitan presentar necesariamente el mismo número de píxeles en la dirección horizontal y en la dirección vertical. Por ejemplo, los bloques pueden comprender $N \times M$ píxeles, donde M no es necesariamente igual a N.

Tras la codificación de intrapredicción o interpredicción mediante las PU de una CU, el codificador de vídeo 20 puede calcular datos residuales para las TU de la CU. Las PU pueden comprender datos sintácticos que describen un procedimiento o modo de generación de datos de píxeles predictivos en el dominio espacial (también denominado como el dominio de píxeles) y las TU pueden comprender coeficientes en el dominio de transformación, tras la aplicación de una transformación, por ejemplo, una transformación de coseno discreta (DCT), una transformación entera, una transformación de ondículas o una transformación conceptualmente similar, a los datos de vídeo residuales. Los datos residuales pueden corresponder a diferencias de píxeles entre píxeles de la imagen no codificada y valores de predicción correspondientes a las PU. El codificador de vídeo 20 puede formar las TU incluyendo los datos residuales para la CU, y a continuación transformar las TU para generar coeficientes de transformada para la CU.

Tras cualquier transformada para generar coeficientes de transformada, el codificador de vídeo 20 puede realizar la cuantización de los coeficientes de transformada. La cuantización se refiere en general a un proceso en el que los coeficientes de transformada se cuantizan para reducir posiblemente la cantidad de datos usados para representar los coeficientes, proporcionando una compresión adicional. El proceso de cuantización puede reducir la profundidad de bits asociada a algunos o la totalidad de los coeficientes. Por ejemplo, un valor de n bits puede redondearse por lo bajo a un valor de m bits durante la cuantización, donde n es mayor que m .

Después de la cuantización, el codificador de vídeo puede escanear los coeficientes de transformación, produciendo un vector unidimensional a partir de la matriz bidimensional que incluye los coeficientes de transformada cuantizados. La exploración puede estar diseñada para colocar coeficientes de energía más alta (y por lo tanto de menor frecuencia) en la parte frontal de la matriz y para colocar coeficientes de energía más bajos (y por lo tanto de mayor frecuencia) en la parte posterior de la matriz. En algunos ejemplos, el codificador de vídeo 20 puede usar un orden de exploración predefinido para explorar los coeficientes de transformada cuantizados y generar un vector en

serie que pueda someterse a la codificación por entropía. En otros ejemplos, el codificador de vídeo 20 puede realizar una exploración adaptativa. Después de explorar los coeficientes de transformada cuantizados para formar un vector unidimensional, el codificador de vídeo 20 puede realizar la codificación por entropía del vector unidimensional, por ejemplo, de acuerdo con la codificación de longitud variable adaptativa de acuerdo con el contexto (CAVLC), la codificación aritmética binaria adaptativa de acuerdo con el contexto (CABAC), la codificación aritmética binaria adaptativa de acuerdo con el contexto basada en la sintaxis (SBAC), la codificación por entropía por división de intervalos de probabilidad (PIPE) u otros procedimientos de codificación por entropía. El codificador de vídeo 20 también puede realizar la codificación por entropía de elementos sintácticos asociados a los datos de vídeo codificados para su uso por el descodificador de vídeo 30 en la descodificación de los datos de vídeo.

Para realizar la CABAC, el codificador de vídeo 20 puede asignar un contexto de un modelo contextual a un símbolo que se va a transmitir. El contexto puede referirse, por ejemplo, a si los valores contiguos del símbolo son distintos de cero o no. Para realizar la CAVLC, el codificador de vídeo 20 puede seleccionar un código de longitud variable para un símbolo que se va a transmitir. Las palabras de código en la VLC pueden construirse de forma que los códigos relativamente más cortos correspondan a símbolos más probables, mientras que los códigos más largos correspondan a símbolos menos probables. De esta manera, el uso de la VLC puede permitir un ahorro en bits con respecto, por ejemplo, al uso de palabras de código de igual longitud para cada símbolo que se va a transmitir. La determinación de la probabilidad puede basarse en un contexto asignado al símbolo.

De acuerdo con las técnicas de esta divulgación, se puede configurar un codificador de vídeo, como un codificador de vídeo 20 o un descodificador de vídeo 30, para codificar un conjunto de parámetros de vídeo (VPS) en una o más capas de datos de vídeo y para codificar una o más capas de datos de vídeo basándose, al menos en parte, en el VPS. Las tablas 2 y 5, descritas con más detalle a continuación, incluyen conjuntos de ejemplos de elementos sintácticos de un VPS. Cada una, o más, de las capas de datos de vídeo puede referirse al VPS, es decir, al mismo VPS. En otras palabras, el VPS puede aplicarse a todas las capas de un conjunto común de datos de vídeo, por ejemplo, todas las capas SVC y/o todas las vistas de datos de vídeo MVC.

El VPS puede incluir diversas categorías de información. Por ejemplo, el VPS puede incluir una descripción de contador de dimensión de muestra (SDCD). Es decir, para cada dimensión, el codificador de vídeo puede señalar un conjunto de índices. Las dimensiones posibles incluyen cnt_p: número de capas prioritarias contenidas en la secuencia de vídeo codificada; cnt_d: cuántas capas de dependencia diferentes en el flujo de bits, múltiples capas con la misma resolución espacial y profundidad de bits pueden pertenecer a diferentes capas de dependencia; cnt_t: cuántas capas temporales en el flujo de bits; cnt_q: número máximo de capas de calidad para cualquier capa de dependencia en el flujo de bits; y cnt_v: número máximo de vistas. Los ajustes de profundidad de bits pueden incluir 8 bits o 12 bits y pueden ser diferentes para diferentes componentes de color. Los formatos de muestreo de croma pueden incluir 4:0:0, 4:2:0 y 4:4:4.

El VPS también puede incluir un índice de muestra para la asignación de características. Si para cada dimensión el indicador de características no es igual a un índice que varía de 0 al contador de dimensiones de muestra menos 1, se puede introducir un bucle que especifica el indicador de características para cada índice de características. La asignación puede incluir, para cada índice de dependencia, una resolución espacial específica con un valor de profundidad de bits específico y un formato de muestra cromática específico. Obsérvese que esto puede omitirse si siempre hay una tabla fija de búsqueda en el descodificador, por ejemplo, 0 puede corresponder a 4:2:0, 1 puede corresponder a 4:4:4 y 2 puede corresponder a 4:0:0. La asignación puede incluir adicionalmente, o de forma alternativa: para cada id / índice temporal, una velocidad de tramas específica o velocidad de tramas media; para cada índice de vista, un id de vista específico; para cada índice de profundidad de bits, un par de valores de profundidad de bits específicos para luma y croma; y para cada formato de muestreo de croma, un indicador de formato de muestreo de croma específico.

El VPS también puede incluir parámetros de control e indicadores de habilitación / deshabilitación de herramientas, tales como los siguientes: un pcm_bit_depth_luma_minus1, un pcm_bit_depth_chroma_minus1, un loop_filter_across_slice_flag, un pcm_loop_filter_disable_flag, un temporal_id_nesting_flag, uno o más elementos sintácticos relacionados con componentes, un chroma_pred_from_luma_enabled_flag, un sample_adaptive_offset_enabled_flag, un adaptive_loop_filter_enabled_flag, y un inter_4x4_enabled_flag.

El VPS también puede incluir una o más descripciones de punto de operación. Los puntos de operación en general describen un subconjunto de un número total de vistas de datos de vídeo incluidos en un flujo de bits. Un punto de operación puede incluir un número particular de vistas dirigidas a la salida, así como otras vistas que se pueden usar como referencia cuando se descodifican, se emiten o ambas cosas. Un flujo de bits puede incluir uno o más puntos de operación descritos por las descripciones de punto de operación. Las descripciones de punto de operación pueden incluir información que define un cierto número de puntos de operación máximos, dependencia entre capas o vistas diferentes, perfil y nivel para cada punto de operación, velocidad de bits para cada punto de operación, dependencia entre puntos de operación, para cada punto de operación, otras restricciones, para cada punto de operación, información de usabilidad de vídeo (VUI) o parte de VUI, y/o para cada capa o vista, VUI o parte de VUI. Además, o de forma alternativa, las descripciones de punto de operación pueden incluir, para cada punto de operación, una representación de unidad de capa de abstracción de red (NAL) de capa de codificación de vídeo

(VCL) de punto de operación. En algunos ejemplos, la representación de unidad VCL NAL de punto de operación puede incluir, para cada dimensión, tres opciones posibles: (1) un valor de índice específico: por ejemplo, para la resolución espacial, para la profundidad de bits en el formato de muestreo de croma; (2) un rango del valor del índice: por ejemplo, para capas temporales, de 0 al id de capa temporal más alta, para capas de calidad, de 0 al id de capa de calidad más alta; o (3) una lista de valores de índice, por ejemplo, para vistas, una lista de valores de índice de vista.

En algunos ejemplos, el VPS puede incluir datos indicativos de un número máximo de capas temporales entre capas de un flujo de bits. Es decir, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden configurarse para codificar un VPS que incluye datos indicativos de un número máximo de capas temporales para un flujo de bits correspondiente. Por ejemplo, el codificador de vídeo 20 puede determinar un número máximo de capas temporales y codificar el VPS para incluir datos que representan el número máximo determinado de capas temporales, mientras que el descodificador de vídeo 30 puede descodificar el VPS para determinar el número máximo de capas temporales. El codificador de vídeo 20 y el descodificador de vídeo 30 también pueden codificar datos de vídeo del flujo de bits basándose en el número máximo determinado de capas temporales. Por ejemplo, el número máximo de capas temporales puede influir en un cierto número de identificadores temporales que son necesarios para representar las diversas capas temporales. Para mencionar otro ejemplo, el número máximo de capas temporales puede influir en la manera en la que el codificador de vídeo 20 y el descodificador de vídeo 30 codifican identificadores de una imagen de referencia, por ejemplo, usando valores de conteo de orden de imagen (POC).

Como otro ejemplo más, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden configurarse para codificar datos de una capa temporal particular usando solamente datos de referencia actuales e incluyendo la misma capa temporal. En otras palabras, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden estar configurados para evitar codificar datos de una capa temporal particular usando los datos de referencia de una capa temporal superior. De esta manera, se puede asegurar que el descodificador de vídeo 30 descodifica con exactitud datos de vídeo de un conjunto dado de capas temporales incluso después de la extracción de flujo de bits secundario. Es decir, si se realiza extracción de flujo de bits secundario, ciertas capas temporales por encima de la capa más alta del flujo de bits secundario extraído no estarán disponibles como referencia. Mediante la codificación de datos de cada capa temporal solo con referencia a datos de capas en, o por debajo, de la capa actual se pueden evitar errores que de otro modo podrían ocurrir como consecuencia de tener datos en una capa particular dependiendo de datos de una capa superior, los cuales se perderían como resultado de la extracción de flujo de bits secundario.

En algunos ejemplos, el VPS, adicionalmente o de forma alternativa, incluye datos indicativos, de cualquiera o ambos, entre un cierto número de imágenes a reordenar en una o más capas de un flujo de bits y/o un cierto número de imágenes a almacenar en una memoria intermedia de imágenes descodificadas (DPB). Como se ha indicado anteriormente, tales datos pueden denominarse información de restricción del flujo de bits. De acuerdo con esto, el dispositivo de destino 14 puede determinar las capacidades del descodificador de vídeo 30 y utilizar la información de restricción del flujo de bits para determinar si el flujo de bits correspondiente es apropiado para ser descodificado por el descodificador de vídeo 30, o si el dispositivo de destino 14 debería seleccionar contenido alternativo (por ejemplo, de un proveedor de contenido basado en red, suponiendo que están disponibles versiones múltiples del contenido).

Además, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden utilizar la información de restricción del flujo de bits durante la codificación de los datos de vídeo. Por ejemplo, el codificador de vídeo 20 puede garantizar que no se viola la información de restricción del flujo de bits. Es decir, suponiendo que la información de restricción del flujo de bits indica que al menos N imágenes deben almacenarse en una DPB, el codificador de vídeo 20 puede asegurar que no más de N imágenes están incluidas en cualquier combinación de una o más listas de imágenes de referencia en un momento dado. Para mencionar otro ejemplo, suponiendo que la información de reordenación de imagen indica que una imagen debe ser desplazada por, al menos, M imágenes, el codificador de vídeo 20 puede asegurar que ninguna imagen se desplace más de M imágenes. El cambio de imágenes de esta manera, en general, se corresponde con la diferencia entre el orden de descodificación y el orden de visualización de una imagen. El descodificador de vídeo 30, asimismo, puede utilizar dicha información durante la codificación, por ejemplo, para realizar la gestión DPB, tal como el lavado con DPB. El codificador de vídeo 20 y el descodificador de vídeo 30 pueden también utilizar información de restricción del flujo de bits, tal como el número máximo de imágenes que se van a almacenar en el DPB y/o el número de imágenes a reordenar, al codificar los valores de identificador de la imagen de referencia.

En algunos ejemplos, el VPS, adicionalmente o de forma alternativa, incluye datos indicativos de parámetros de descodificador de referencia hipotéticos (HRD). Los parámetros HRD incluyen, por ejemplo, datos que describen tiempos en los que se deben eliminar datos de una memoria intermedia de imágenes codificadas (CPB). En los descodificadores, tales como el descodificador de vídeo 30, la CPB representa una memoria intermedia en la que se almacenan datos de vídeo codificados hasta que los datos están listos para la descodificación. Los descodificadores tales como el descodificador de vídeo 30 pueden incluir también una memoria intermedia de imágenes descodificadas (DPB), en el que se almacenan datos de vídeo descodificados, por ejemplo, para usarse como datos de referencia de datos interpredichos y para reordenar imágenes de un orden de descodificación a un orden de

visualización.

Los parámetros HRD pueden incluir datos que indican cuándo se deben eliminar las imágenes particulares de la CPB y descodificarlas. De este modo, el codificador de vídeo 20 puede codificar los parámetros de HRD del VPS para indicar cuándo las imágenes pueden ser eliminadas de la CPB y descodificadas, mientras que el descodificador de vídeo 30 puede descodificar los parámetros de HRD del VPS para determinar cuándo eliminar las imágenes de la CPB. Del mismo modo, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden codificar imágenes de acuerdo con los parámetros de HRD, por ejemplo, en un orden de codificación indicado por los parámetros de HRD. De esta manera, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden configurarse para codificar un VPS que incluye parámetros de HRD y para codificar datos de vídeo correspondientes a los VPS basándose, al menos en parte, en los parámetros de HRD.

El VPS también puede incluir datos de ampliación que indican si el VPS se ha ampliado, por ejemplo, para proporcionar datos de una o más herramientas de codificación adicionales. Dichas herramientas de codificación pueden ser herramientas que son diferentes de las de una norma de codificación de vídeo correspondiente, tal como, por ejemplo, ITU-T H.264/AVC o la próxima norma HEVC. Además, tales herramientas de codificación pueden requerir datos de configuración. Estos datos de configuración pueden proporcionarse en los datos de ampliación de un VPS. De esta manera, al codificar datos de vídeo usando tales herramientas de codificación, el codificador de vídeo 20 y/o descodificador de vídeo 30 pueden codificar un VPS que indica si hay datos de ampliación y, si es así, datos de ampliación del VPS. Además, cuando tales datos de ampliación están presentes, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden ejecutar las herramientas de codificación correspondientes para codificar los datos de vídeo usando los datos de ampliación.

Varias normas de codificación de vídeo definen la sintaxis, la semántica y el proceso de descodificación correspondientes a los flujos de bits libres de errores, cualquiera de los cuales se ajustan a cierto perfil o nivel. Las normas de codificación de vídeo en general no especifican el codificador, pero el codificador tiene la tarea de garantizar que los flujos de bits generados sean compatibles con un descodificador. En el contexto de las normas de codificación de vídeo, un "perfil" corresponde a un subconjunto de algoritmos, características o herramientas y restricciones que se les aplican. Como se define en la norma H.264, por ejemplo, un "perfil" es un subconjunto de toda la sintaxis del flujo de bits especificada por la norma H.264. Un "nivel" corresponde a las limitaciones del consumo de recursos del descodificador, tales como, por ejemplo, memoria de descodificador y cálculo, que están relacionadas con la resolución de las imágenes, velocidad de bits y velocidad de procesamiento de bloques. Un perfil puede ser señalado con un valor `profile_idc` (indicador de perfil), mientras que un nivel puede ser señalado con un valor `level_idc` (indicador de nivel). De acuerdo con las técnicas de esta divulgación, la información de perfil y nivel se puede especificar en las descripciones de punto de operación, tal como se ha analizado anteriormente.

En algunos ejemplos, cada capa o vista de un flujo de bits se refiere al conjunto de parámetros de vídeo (VPS), y un conjunto de parámetros de secuencia de capas (LPS) puede estar activo para cada capa. Un LPS puede mantenerse lo más ligero posible haciendo referencia al VPS en el diseño. El LPS puede incluir cualquiera o toda la información analizada a continuación. El LPS puede incluir una indicación de dimensión de muestra que indica, para cada dimensión, un índice para cada dimensión. Por ejemplo, si en un VPS, se asigna un índice de resolución espacial 0 a una característica espacial de 320x240 y se asigna un índice de resolución espacial 1 a 640x480, y la capa actual se va a asignar con una resolución de 640x480, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden codificar un elemento sintáctico con un valor de 1 para la capa actual. Es decir, el codificador de vídeo 20 puede indicar un valor de 1 para que el elemento sintáctico especifique la resolución de 640 x 480, mientras que el descodificador de vídeo 30 puede determinar que, una capa actual con un elemento sintáctico con un valor de 1, tiene una resolución de 640 x 480, basándose en el valor de 1 del elemento sintáctico.

El LPS también puede incluir parámetros de control e indicadores de habilitación / deshabilitación de herramientas. Por ejemplo, los parámetros de control e indicadores de habilitación / deshabilitación de herramientas pueden incluir un `pcm_bit_depth_luma_minus1`, un `pcm_bit_depth_chroma_minus1`, un `loop_filter_across_slice_flag`, un `pcm_loop_filter_disable_flag`, uno o más elementos sintácticos relacionados con componentes, un `chroma_pred_from_luma_enabled_flag`, un `sample_adaptive_offset_enabled_flag`, un `adaptive_loop_filter_enabled_flag`, y una jerarquía de unidad de codificación (CU).

El LPS puede incluir además información de otros tipos de conjuntos de parámetros aplicables a un fragmento, un grupo de fragmentos, una imagen o varias imágenes. Cada uno de estos conjuntos de parámetros puede referirse a un conjunto de parámetros de imagen específico (PPS).

El codificador de vídeo, tal como el codificador de vídeo 20 y el descodificador de vídeo 30, pueden configurarse para asegurar y/o determinar que un PPS no haga referencia a un LPS o un VPS. Por lo tanto, el codificador de vídeo puede garantizar que cada PPS de un flujo de bits no se refiere a un LPS o un VPS. El análisis de un PPS puede ser independiente. Cuando un PPS incluye uno o más de los mismos elementos sintácticos que los de un VPS o un LPS, los elementos sintácticos del PPS pueden sobrescribir los del VPS o LPS.

Además, se puede configurar un codificador de vídeo para codificar un conjunto de parámetros de agrupación (GPS)

que agrupa todos los conjuntos de parámetros. El codificador de vídeo puede codificar una pluralidad de diferentes grupos dentro del GPS, cada uno de los cuales tiene identificadores (ids) de GPS individuales. Cada uno de los grupos en el GPS puede incluir una combinación diferente de conjuntos de parámetros. De esta manera, una cabecera de fragmento solo tiene que incluir una referencia a un id de GPS correspondiente y no necesita incluir una indicación de un tipo de conjunto de parámetros. La solicitud de patente provisional estadounidense número de serie 61/590.702, presentada el 25 de enero de 2012, también describe técnicas en las que se agrupan diferentes tipos de conjuntos de parámetros y solo el ID del RBSP de agrupación de conjunto de parámetros se señala en la cabecera del fragmento en mayor detalle.

Como se ha analizado anteriormente, el codificador de vídeo, tal como el codificador de vídeo 20 o el decodificador de vídeo 30, puede configurarse para codificar un conjunto de parámetros de vídeo y/o un conjunto de parámetros de agrupación. Los ejemplos de un conjunto de parámetros de vídeo se analizan con mayor detalle con respecto a la FIG. 5, mientras que los ejemplos de un conjunto de parámetros de agrupación se analizan con mayor detalle con respecto a la FIG. 6.

Además, el codificador de vídeo 20 puede enviar datos sintácticos, tales como datos sintácticos basados en bloques, datos sintácticos basados en tramas y datos sintácticos basados en GOP, al decodificador de vídeo 30, por ejemplo, en una cabecera de trama, una cabecera de bloque, una cabecera de fragmento o una cabecera de GOP. Los datos sintácticos de GOP pueden describir un cierto número de tramas en el GOP respectivo, y los datos sintácticos de trama pueden indicar un modo de codificación/predicción usado para codificar la trama correspondiente.

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

La FIG. 2 es un diagrama de bloques que ilustra un ejemplo del codificador de vídeo 20 que puede implementar técnicas para la codificación de conjuntos de parámetros y unidades NAL para una o más capas de datos de vídeo. El codificador de vídeo 20 puede realizar intracodificación e intercodificación de bloques de vídeo dentro de fragmentos de vídeo. La intracodificación se basa en la predicción espacial para reducir o eliminar la redundancia espacial en el vídeo dentro de una trama o imagen de vídeo dada. La intercodificación se basa en la predicción temporal para reducir o eliminar la redundancia temporal en el vídeo dentro de tramas o imágenes adyacentes de una secuencia de vídeo. La intramodalidad (modalidad I) puede referirse a cualquiera de varias modalidades de compresión de base espacial. Los intermodos, tales como la predicción unidireccional (modo P) o la bi-predicción (modo B), pueden referirse a cualquiera de varios modos de codificación de base temporal.

Como se muestra en la FIG. 2, el codificador de vídeo 20 recibe un bloque de vídeo actual dentro de una trama de vídeo a codificar. En el ejemplo de la FIG. 2, el codificador de vídeo 20 incluye una unidad de selección de modo 40, una memoria de imágenes de referencia 64, un sumador 50, una unidad de procesamiento de transformación 52, una unidad de cuantización 54 y una unidad de codificación por entropía 56. A su vez, la unidad de selección de modo 40 incluye una unidad de compensación de movimiento 44, una unidad de estimación de movimiento 42, una unidad de intrapredicción 46 y una unidad de división 48. Para la reconstrucción de bloques de vídeo, el codificador de vídeo 20 incluye además una unidad de cuantización inversa 58, una unidad de transformación inversa 60 y un sumador 62. También puede incluirse un filtro de desbloqueo (no se muestra en la FIG. 2) para filtrar límites de bloque, para eliminar distorsiones de efecto pixelado del vídeo reconstruido. Si se desea, el filtro de desbloqueo filtrará típicamente la salida del sumador 62. También pueden usarse filtros adicionales (en bucle o pos-bucle), además del filtro de desbloqueo. Dichos filtros no se muestran por razones de brevedad pero, si se desea, pueden filtrar la salida del sumador 50 (tal como un filtro en bucle).

Durante el proceso de codificación, el codificador de vídeo 20 recibe una trama o un fragmento de vídeo que va a codificarse. La trama o el fragmento puede estar dividido en múltiples bloques de vídeo. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 llevan a cabo la codificación interpredictiva del bloque de vídeo recibido, con respecto a uno o más bloques de una o más tramas de referencia, para proporcionar la predicción temporal. La unidad de intrapredicción 46, como alternativa, puede llevar a cabo la codificación intrapredictiva del bloque de vídeo recibido, con respecto a uno o más bloques contiguos de la misma trama o fragmento que el bloque que va a codificarse, para proporcionar la predicción espacial. El codificador de vídeo 20 puede llevar a cabo múltiples pasadas de codificación, por ejemplo, para seleccionar un modo de codificación adecuada para cada bloque de datos de vídeo.

Además, la unidad de división 48 puede dividir bloques de datos de vídeo en sub-bloques, basándose en la

evaluación de los anteriores esquemas de división en las pasadas de codificación anteriores. Por ejemplo, la unidad de división 48 puede dividir inicialmente una trama o un fragmento en varias LCU, y dividir cada una de las LCU en varias sub-CU, basándose en un análisis de distorsión de velocidad (por ejemplo, optimización de distorsión de velocidad). La unidad de selección de modo 40 puede producir además una estructura de datos de árbol cuádruple que indica la división de una LCU en varias sub-CU. Las CU de nodos de hojas del árbol cuádruple pueden incluir una o más PU y una o más TU.

La unidad de selección de modo 40 puede seleccionar uno de los modos de codificación, intra o inter, por ejemplo, en función de los resultados de error, y proporciona el bloque intracodificado o intercodificado resultante al sumador 50 para generar datos de bloque residuales y al sumador 62 para reconstruir el bloque codificado para usarse como una trama de referencia. La unidad de selección de modalidad 40 proporciona además elementos sintácticos, tales como vectores de movimiento, indicadores de intramodalidad, información de división y otra información sintáctica de este tipo, a la unidad de codificación por entropía 56.

La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden estar sumamente integradas, pero se ilustran por separado con fines conceptuales. La estimación de movimiento, realizada por la unidad de estimación de movimiento 42, es el proceso de generación de vectores de movimiento, que estiman el movimiento de los bloques de vídeo. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de una PU de un bloque de vídeo dentro de una trama o imagen actual de vídeo, con respecto a un bloque predictivo dentro de una trama de referencia (u otra unidad codificada), con respecto al bloque actual que está siendo codificado dentro de la trama actual (u otra unidad codificada). Un bloque predictivo es un bloque que se revela como estrechamente coincidente con el bloque a codificar en lo que respecta a la diferencia de píxeles, que puede determinarse mediante la suma de una diferencia absoluta (SAD), una suma de diferencia de cuadrados (SSD) u otras métricas de diferencia. En algunos ejemplos, el codificador de vídeo 20 puede calcular valores de posiciones de píxel subentero de las imágenes de referencia almacenadas en la memoria de imágenes de referencia 64. Por ejemplo, el codificador de vídeo 20 puede interpolar valores de posiciones de un cuarto de píxel, posiciones de un octavo de píxel u otras posiciones de píxel fraccionarias de la imagen de referencia. Por lo tanto, la unidad de estimación de movimiento 42 puede realizar una búsqueda de movimiento con respecto a las posiciones de píxel completo y a las posiciones de píxel fraccionario, y emitir un vector de movimiento con una precisión de píxel fraccionario.

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

La compensación de movimiento, llevada a cabo por la unidad de compensación de movimiento 44, puede implicar capturar o generar el bloque predictivo basándose en el vector de movimiento determinado por la unidad de estimación de movimiento 42. De nuevo, la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden integrarse funcionalmente, en algunos ejemplos. Tras recibir el vector de movimiento para la PU del bloque de vídeo actual, la unidad de compensación de movimiento 44 puede localizar el bloque predictivo al que apunta el vector de movimiento en una de las listas de imágenes de referencia. El sumador 50 forma un bloque de vídeo residual restando los valores de píxel del bloque predictivo a los valores de píxel del bloque de vídeo actual que está siendo codificado, generando valores de diferencia de píxel, como se analiza posteriormente. En general, la unidad de estimación de movimiento 42 lleva a cabo la estimación de movimiento con respecto a los componentes de luma, y la unidad de compensación de movimiento 44 utiliza los vectores de movimiento calculados basándose en los componentes de luma, tanto para los componentes de croma como para los componentes de luma. La unidad de selección de modalidad 40 también puede generar elementos sintácticos asociados a los bloques de vídeo y al fragmento de vídeo para su uso por parte del decodificador de vídeo 30 a la hora de decodificar los bloques de vídeo del fragmento de vídeo.

La unidad de intrapredicción 46 puede intrapredicir un bloque actual, como alternativa a la interpredicción llevada a cabo por la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44, como se ha descrito anteriormente. En particular, la unidad de intrapredicción 46 puede determinar una modalidad de intrapredicción a usar para codificar un bloque actual. En algunos ejemplos, la unidad de intrapredicción 46 puede codificar un bloque actual usando varias modalidades de intrapredicción, por ejemplo, durante diferentes pasadas de codificación, y la unidad de intrapredicción 46 (o la unidad de selección de modalidad 40, en algunos ejemplos) puede seleccionar una modalidad adecuada de intrapredicción a usar entre las modalidades probadas.

Por ejemplo, la unidad de intrapredicción 46 puede calcular valores de velocidad-distorsión usando un análisis de velocidad-distorsión para las diversas modalidades de intrapredicción probadas, y seleccionar la modalidad de intrapredicción que tenga las mejores características de velocidad-distorsión entre las modalidades probadas. El análisis de velocidad-distorsión determina en general una magnitud de distorsión (o errores) entre un bloque

codificado y un bloque original, no codificado, que se codificó para generar el bloque codificado, así como una velocidad binaria (es decir, un cierto número de bits) usada para generar el bloque codificado. La unidad de intrapredicción 46 puede calcular proporciones a partir de las distorsiones y velocidades de los diversos bloques codificados para determinar qué modalidad de intrapredicción presenta el mejor valor de velocidad-distorsión para el bloque.

Después de seleccionar una modalidad de intrapredicción para un bloque, la unidad de intrapredicción 46 puede proporcionar información, indicativa de la modalidad de intrapredicción seleccionada para el bloque, a la unidad de codificación por entropía 56. La unidad de codificación por entropía 56 puede codificar la información que indica la modalidad de intrapredicción seleccionada. El codificador de vídeo 20 puede incluir datos de configuración de flujo de bits transmitido, que pueden incluir una pluralidad de tablas de índices de modalidades de intrapredicción y una pluralidad de tablas de índices de modalidades de intrapredicción modificadas (también denominadas tablas de asignación de palabras de código), definiciones de contextos de codificación para varios bloques e indicaciones de una modalidad de intrapredicción más probable, una tabla de índices de modalidades de intrapredicción y una tabla modificada de índices de modalidades de intrapredicción, a usar en cada uno de los contextos.

El codificador de vídeo 20 forma un bloque de vídeo residual restando los datos de predicción de la unidad de selección de modalidad 40 a partir del bloque de vídeo original que está siendo codificado. El sumador 50 representa el componente o los componentes que realizan esta operación de resta. La unidad de procesamiento de transformaciones 52 aplica una transformación, tal como una transformación discreta del coseno (DCT) o una transformación conceptualmente similar al bloque residual, generando un bloque de vídeo que comprende valores residuales de coeficientes de transformación. La unidad de procesamiento de transformaciones 52 puede llevar a cabo otras transformaciones que son conceptualmente similares a la DCT. También podrían usarse transformaciones de ondículas, transformaciones de enteros, transformaciones de subbandas u otros tipos de transformaciones.

En cualquier caso, la unidad de procesamiento de transformaciones 52 aplica la transformación al bloque residual, generando un bloque de coeficientes de transformación residuales. La transformación puede convertir la información residual, desde un dominio de valores de píxel a un dominio de transformaciones, como un dominio de frecuencia. La unidad de procesamiento de transformaciones 52 puede enviar los coeficientes de transformación resultantes a la unidad de cuantización 54. La unidad de cuantización 54 cuantiza los coeficientes de transformación para reducir adicionalmente la velocidad de bits. El proceso de cuantización puede reducir la profundidad de bits asociada a algunos de, o a todos, los coeficientes. El grado de cuantización puede modificarse ajustando un parámetro de cuantización. En algunos ejemplos, la unidad de cuantización 54 puede realizar, a continuación, una exploración de la matriz que incluye los coeficientes de transformada cuantizados. De forma alternativa, la unidad de codificación por entropía 56 puede realizar la exploración.

Tras la cuantización, la unidad de codificación por entropía 56 codifica por entropía los coeficientes de transformación cuantizados. Por ejemplo, la unidad de codificación por entropía 56 puede llevar a cabo la codificación de longitud variable adaptativa al contexto (CAVLC), la codificación aritmética binaria adaptativa al contexto (CABAC), la codificación aritmética binaria adaptativa al contexto y basada en sintaxis (SBAC), la codificación por entropía mediante la división en intervalos de probabilidades (PIPE) u otra técnica de codificación por entropía. En el caso de la codificación por entropía basada en el contexto, el contexto puede basarse en bloques contiguos. Tras la codificación por entropía realizada por la unidad de codificación por entropía 56, el flujo de bits codificado puede transmitirse a otro dispositivo (por ejemplo, el descodificador de vídeo 30) o archivar para su posterior transmisión o recuperación.

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

El codificador de vídeo 20 puede configurarse adicionalmente para codificar un conjunto de parámetros de vídeo (VPS), un conjunto de parámetros de capa (LPS) y/o un conjunto de parámetros de agrupación, de acuerdo con las técnicas de esta divulgación, así como un conjunto de parámetros de secuencia SPS), un conjunto de parámetros de imagen (PPS), un conjunto de parámetros de adaptación (APS) u otras estructuras de datos de señalización de este tipo. Más particularmente, la unidad de codificación por entropía 56 puede estar configurada para codificar cualquiera de, o todas, estas estructuras de datos. En la medida en que los parámetros de estas diversas estructuras de datos pueden afectar al rendimiento de codificación, la unidad de selección de modo 40 puede

seleccionar parámetros apropiados y pasar los parámetros a la unidad de codificación por entropía 56 para su inclusión dentro, por ejemplo, de un VPS. Un usuario, por ejemplo un administrador, puede seleccionar otros parámetros, tales como un cierto número de capas temporales, un cierto número de imágenes a reordenar y un cierto número de imágenes a almacenar en una memoria intermedia de imágenes descodificadas. En otros ejemplos, ciertos parámetros, como los parámetros HRD, pueden surgir a través del proceso de codificación.

La unidad de codificación por entropía 56 puede codificar un VPS para incluir cualquiera de, o todos, los diversos tipos de datos descritos en esta divulgación. El codificador de vídeo 20 también puede codificar datos de acuerdo con los parámetros del VPS. Más en particular, el codificador de vídeo 20 puede codificar secuencias de imágenes entre una o más capas de datos de vídeo a las que corresponde el VPS de acuerdo con los parámetros del VPS.

De esta manera, el codificador de vídeo 20 de la FIG. 2 representa un ejemplo de un codificador de vídeo configurado para codificar un conjunto de parámetros de vídeo (VPS) para una o más capas de datos de vídeo, en el que cada una de una o más capas de datos de vídeo se refiere al VPS, y codificar una o más capas de datos de vídeo basándose, al menos en parte, en el VPS.

Aunque en general se describe con respecto a un codificador de vídeo, la codificación de un VPS puede ser realizada por otros dispositivos, por ejemplo, un elemento de red de soporte de medios (MANE). Un MANE puede corresponder a un elemento de red entre un dispositivo fuente (como el dispositivo fuente 12 de la FIG. 1) y un dispositivo de destino (como el dispositivo de destino 14). El MANE puede configurarse para codificar un VPS de acuerdo con las técnicas de esta divulgación. El MANE puede generar el VPS utilizando datos de otras estructuras de datos recibidas por el MANE, por ejemplo, conjuntos de parámetros de secuencia.

La FIG. 3 es un diagrama de bloques que ilustra un ejemplo del descodificador de vídeo 30 que puede implementar las técnicas de codificación de conjuntos de parámetros y unidades NAL para una o más capas de datos de vídeo. En el ejemplo de la FIG. 3, el descodificador de vídeo 30 incluye una unidad de descodificación por entropía 70, una unidad de compensación de movimiento 72, una unidad de intrapredicción 74, una unidad de cuantización inversa 76, una unidad transformada inversa 78, una memoria de imágenes de referencia 82 y un sumador 80. La memoria de imágenes de referencia 82 también se puede denominar "memoria intermedia de imágenes descodificadas" o DPB. En algunos ejemplos, el descodificador de vídeo 30 puede llevar a cabo una pasada de descodificación en general recíproca a la pasada de codificación descrita con respecto al codificador de vídeo 20 (FIG. 2). La unidad de compensación de movimiento 72 puede generar datos de predicción basados en vectores de movimiento recibidos desde la unidad de descodificación por entropía 70, mientras que la unidad de intrapredicción 74 puede generar datos de predicción basados en indicadores de modalidad de intrapredicción recibidos desde la unidad de descodificación por entropía 70.

Durante el proceso de descodificación, el descodificador de vídeo 30 recibe un flujo de bits de vídeo codificado que representa bloques de vídeo de un fragmento de vídeo codificado y elementos sintácticos asociados desde el codificador de vídeo 20. La unidad de descodificación por entropía 70 del descodificador de vídeo 30 descodifica por entropía el flujo de bits para generar coeficientes cuantizados, vectores de movimiento o indicadores de modalidad de intrapredicción y otros elementos sintácticos. La unidad de descodificación por entropía 70 remite los vectores de movimiento y otros elementos sintácticos a la unidad de compensación de movimiento 72. El descodificador de vídeo 30 puede recibir los elementos sintácticos al nivel del fragmento de vídeo y/o el nivel del bloque de vídeo.

Cuando el fragmento de vídeo se codifica como un fragmento intracodificado (I), la unidad de intrapredicción 74 puede generar datos de predicción para un bloque de vídeo del fragmento de vídeo actual, basándose en una modalidad de intrapredicción señalada, y datos de bloques previamente descodificados de la trama o imagen actual. Cuando la trama de vídeo se codifica como un fragmento intercodificado (es decir, B, P o GPB), la unidad de compensación de movimiento 72 genera bloques predictivos para un bloque de vídeo del fragmento de vídeo actual, basándose en los vectores de movimiento y a otros elementos sintácticos recibidos desde la unidad de descodificación por entropía 70. Los bloques predictivos pueden ser generados a partir de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. El descodificador de vídeo 30 puede construir las listas de tramas de referencia, la Lista 0 y la Lista 1, usando técnicas de construcción por omisión, basándose en las imágenes de referencia almacenadas en la memoria de imágenes de referencia 82.

La unidad de compensación de movimiento 72 determina la información de predicción para un bloque de vídeo del fragmento de vídeo actual, analizando los vectores de movimiento y otros elementos sintácticos, y usa la información de predicción para generar los bloques predictivos del bloque de vídeo actual que está siendo descodificado. Por ejemplo, la unidad de compensación de movimiento 72 usa algunos de los elementos sintácticos recibidos para determinar un modo de predicción (por ejemplo, intrapredicción o interpredicción), usada 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 del fragmento, vectores de movimiento para cada bloque de vídeo intercodificado del fragmento, el estado de interpredicción para cada bloque de vídeo intercodificado del fragmento y otra información para descodificar los bloques de vídeo del fragmento de vídeo actual.

La unidad de compensación de movimiento 72 también puede realizar la interpolación basándose en filtros de interpolación. La unidad de compensación de movimiento 72 puede usar filtros de interpolación, como los usados por el codificador de vídeo 20 durante la codificación de los bloques de vídeo, para calcular valores interpolados de píxeles subenteros de los bloques de referencia. En este caso, la unidad de compensación de movimiento 72 puede

5 determinar los filtros de interpolación usados por el codificador de vídeo 20 a partir de los elementos sintácticos recibidos y usar los filtros de interpolación para generar bloques predictivos.

La unidad de cuantización inversa 76 cuantiza de manera inversa, es decir, descuantiza, los coeficientes de transformación cuantizados, proporcionados en el flujo de bits y descodificados por la unidad de descodificación por entropía 80. El proceso de cuantización inversa puede incluir el uso de un parámetro de cuantización QPY, calculado por el descodificador de vídeo 30 de cada bloque de vídeo en el fragmento de vídeo para determinar un grado de cuantización y, asimismo, un grado de cuantización inversa que debería aplicarse. La unidad de transformación inversa 78 aplica una transformación inversa, por ejemplo, una DCT inversa, una transformación inversa entera o un proceso de transformación inversa conceptualmente similar a los coeficientes de transformación

15 con el fin de generar bloques residuales en el dominio de píxeles.

Después de que la unidad de compensación de movimiento 72 genera el bloque predictivo del bloque de vídeo actual, basándose en los vectores de movimiento y a otros elementos sintácticos, el descodificador de vídeo 30 forma un bloque de vídeo descodificado sumando los bloques residuales procedentes de la unidad de transformación inversa 78 y los correspondientes bloques predictivos generados por la unidad de compensación de movimiento 72. El sumador 90 representa el componente o los componentes que llevan a cabo esta operación de suma. Si se desea, también puede aplicarse un filtro de desbloqueo para filtrar los bloques descodificados con el fin de eliminar distorsiones de efecto pixelado. También pueden utilizarse otros filtros de bucle (ya sea en el bucle de codificación o después del bucle de codificación) para suavizar las transiciones de píxeles, o mejorar de otro modo la calidad del vídeo. Los bloques de vídeo descodificados de una trama o imagen dada son a continuación almacenados en la memoria de imágenes de referencia 82, que almacena imágenes de referencia usadas para la posterior compensación de movimiento. La memoria de imágenes de referencia 82 almacena también vídeo descodificado para su presentación posterior en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la FIG. 1.

30

De acuerdo con las técnicas de esta divulgación, el descodificador de vídeo 30 puede descodificar un conjunto de parámetros de vídeo (VPS), un conjunto de parámetros de capa (LPS) y/o un conjunto de parámetros de agrupación, de acuerdo con las técnicas de esta divulgación, así como un conjunto de parámetros de secuencia (SPS), un conjunto de parámetros de imagen (PPS), un conjunto de parámetros de adaptación (APS) u otras estructuras de datos de señalización de este tipo. Más particularmente, la unidad de descodificación por entropía 70 puede estar configurada para descodificar cualquiera de, o todas, estas estructuras de datos. Mediante la descodificación de estas diversas estructuras de datos, la unidad 70 de descodificación por entropía puede determinar parámetros que se utilizarán para descodificar datos de vídeo correspondientes. Por ejemplo, el descodificador de vídeo 30 puede descodificar secuencias correspondientes de datos de vídeo de una o más capas utilizando parámetros de un VPS descodificado.

40

Aunque no se muestra en la FIG. 3, el descodificador de vídeo 30 puede incluir adicionalmente una memoria intermedia de imágenes descodificadas (CPB). Normalmente, la CPB se proporcionaría de forma ordinaria antes de la unidad 70 de descodificación por entropía. De forma alternativa, la CPB puede estar acoplada a la unidad de descodificación por entropía 70 para almacenamiento temporal, o a la salida de la unidad de descodificación por entropía 70 para almacenar datos descodificados por entropía hasta que dichos datos se descodifiquen. En general, la CPB almacena datos de vídeo codificados hasta que los datos de vídeo codificados se van a descodificar, por ejemplo, como se indica mediante los parámetros de HRD, que el descodificador de vídeo 30 puede extraer de un VPS descodificado. Del mismo modo, otros elementos del descodificador de vídeo 30 pueden estar configurados para descodificar datos de vídeo utilizando, por ejemplo, el VPS. Por ejemplo, el descodificador de vídeo 30 puede descodificar identificadores temporales de imágenes de varias capas temporales, datos que indican un cierto número de imágenes a reordenar y/o almacenar en la memoria de imágenes de referencia 82 (que representa una DPB).

50

Además, el descodificador de vídeo 30 puede incluir unidades de procesamiento adicionales para procesar datos de vídeo de acuerdo con diversas herramientas de codificación proporcionadas por ampliaciones de una norma de codificación de vídeo. De forma alternativa, los elementos existentes del descodificador de vídeo 30 mostrado en la FIG. 3 pueden configurarse para ejecutar las herramientas de codificación de tales ampliaciones. La unidad de descodificación por entropía 70 puede estar configurada para descodificar datos de ampliación de VPS y proporcionar tales datos de ampliación a las unidades configuradas para ejecutar las herramientas de codificación proporcionadas por las ampliaciones.

60

De esta manera, el descodificador de vídeo 30 de la FIG. 3 representa un ejemplo de un descodificador de vídeo configurado para codificar un conjunto de parámetros de vídeo (VPS) para una o más capas de datos de vídeo, en el que cada una de una o más capas de datos de vídeo se refieren al VPS, y codificar una o más capas de datos de vídeo basándose al menos en parte en el VPS.

65

Aunque en general se describe con respecto a un decodificador de vídeo, la decodificación de un VPS puede ser realizada por otros dispositivos, por ejemplo, un elemento de red que conoce los medios (MANE). El MANE puede configurarse para decodificar un VPS de acuerdo con las técnicas de esta divulgación. El MANE puede generar además otros datos del conjunto de parámetros, tales como uno o más conjuntos de parámetros de secuencia, utilizando los datos del VPS. De esta manera, el MANE puede proporcionar compatibilidad retrospectiva con las normas anteriores, tales como ITU-T H.264/AVC.

La FIG. 4 es un diagrama conceptual que ilustra un patrón de predicción de la norma MVC a modo de ejemplo. La codificación de vídeo multivista (MVC) es una ampliación de ITU-T H.264/AVC. Una técnica similar puede aplicarse a HEVC. En el ejemplo de la FIG. 4, se ilustran ocho vistas (con ID de vista "S0" a "S7") y se ilustran doce posiciones temporales ("T0" a "T11") para cada vista. Es decir, cada fila de la FIG. 4 corresponde a una vista, mientras que cada columna indica una ubicación temporal.

Una estructura de predicción MVC típica (que incluye tanto la predicción entre imágenes dentro de cada vista como la predicción entre vistas) para la codificación de vídeo multivista se muestra en la FIG. 4, donde las predicciones se indican mediante flechas, donde el objeto al que se apunta utiliza el objeto desde el que se apunta como referencia de predicción. En MVC, la predicción entre vistas es compatible con la compensación de movimiento de disparidad, que puede utilizar la sintaxis de la compensación de movimiento H.264/AVC, pero permite utilizar una imagen en una vista diferente como una imagen de referencia.

La codificación de dos vistas también podría ser compatible con MVC, y una de las ventajas de MVC es que un codificador MVC podría tomar más de dos vistas como una entrada de vídeo 3D y un decodificador MVC puede decodificar una representación multivista. Así, cualquier renderizador con decodificador MVC puede estar configurado para recibir contenido de vídeo 3D con más de dos vistas.

Aunque MVC tiene una denominada vista de base que es decodificable mediante los decodificadores H.264/AVC y el par de vistas estéreo podría ser compatible también con MVC, una ventaja de MVC es que podría permitir un ejemplo que usa más de dos vistas como una entrada de vídeo 3D y decodifica este vídeo 3D representado por las múltiples vistas. Un renderizador de un cliente que tiene un decodificador MVC puede esperar contenido de vídeo 3D con varias vistas.

Un orden de decodificación típico de MVC se denomina codificación de primera vez. Una unidad de acceso puede incluir imágenes codificadas de todas las vistas para una instancia de tiempo de salida. Por ejemplo, cada una de las imágenes del tiempo T0 puede estar incluida en una unidad de acceso común, cada una de las imágenes del tiempo T1 puede estar incluida en una segunda unidad de acceso común, y así sucesivamente. El orden de decodificación no es necesariamente idéntico al orden de salida o de visualización.

Las tramas en la FIG. 4 se indican en la intersección de cada fila y cada columna de la FIG. 4 usando un bloque sombreado que incluye una letra, que designa si la trama correspondiente está intracodificada (es decir, una trama I), o intercodificada en una dirección (es decir, como una trama P) o en múltiples direcciones (es decir, como una trama B). En general, las predicciones se indican mediante flechas, donde la trama a la que se apunta utiliza el objeto desde el que se apunta como referencia de predicción. Por ejemplo, la trama P de la vista S2 en la ubicación temporal T0 se predice a partir de la trama I de la vista S0 en la ubicación temporal T0.

Al igual que con la codificación de vídeo de vista única, las tramas de una secuencia de vídeo de codificación de vídeo multivista pueden codificarse predictivamente con respecto a las tramas en diferentes ubicaciones temporales. Por ejemplo, la trama B de la vista S0 en la ubicación temporal T1 tiene una flecha apuntando a la misma desde la trama I de la vista S0 en la ubicación temporal T0, lo cual indica que la trama B se predice a partir de la trama I. Adicionalmente, sin embargo, en el contexto de la codificación de vídeo multivista, las tramas pueden predecirse entre vistas. Es decir, un componente de vista puede utilizar los componentes de vista en otras vistas como referencia. En MVC, por ejemplo, la predicción entre vistas se realiza como si el componente de vista en otra vista es una referencia de interpredicción. Las posibles referencias entre vistas se señalan en la ampliación de la norma MVC del conjunto de parámetros de secuencia (SPS) y pueden ser modificadas por el proceso de construcción de la lista de imágenes de referencia, lo cual habilita el ordenamiento flexible de las referencias de interpredicción o de predicción entre vistas.

En la ampliación de MVC de H.264/AVC, por ejemplo, la predicción entre vistas es compatible con la compensación de movimiento de disparidad, que utiliza la sintaxis de la compensación de movimiento H.264/AVC, pero permite que se utilice una imagen en una vista diferente como imagen de referencia. La codificación de dos vistas puede ser compatible con MVC, lo cual se denomina en general vistas estereoscópicas. Una de las ventajas de MVC es que un codificador MVC podría tomar más de dos vistas como una entrada de vídeo 3D y un decodificador MVC puede decodificar una representación multivista. Por lo tanto, un dispositivo de renderizado con un decodificador MVC puede esperar contenidos de vídeo 3D con más de dos vistas.

En MVC, se permite la predicción entre vistas (IVP) entre imágenes en la misma unidad de acceso (es decir, con la

misma instancia de tiempo). Una unidad de acceso es, en general, una unidad de datos que incluye todos los componentes de vista (por ejemplo, todas las unidades NAL) para una instancia temporal común. Por lo tanto, en MVC, se permite la predicción entre vistas entre imágenes en la misma unidad de acceso. Cuando se codifica una imagen en una de las vistas no básicas, la imagen puede agregarse a una lista de imágenes de referencia, si está en una vista diferente pero con la misma instancia de tiempo (por ejemplo, el mismo valor POC y, por lo tanto, en la misma unidad de acceso). Una imagen de referencia de predicción entre vistas puede ponerse en cualquier posición de una lista de imágenes de referencia, como es el caso con cualquier imagen de referencia de interpredicción.

En el contexto de la codificación de vídeo multivista hay dos tipos de vectores de movimiento. Uno son vectores de movimiento normales que apuntan a las imágenes de referencia temporales, y el correspondiente modo de interpredicción se denomina una predicción compensada por movimiento (MCP). El otro son vectores de movimiento de disparidad que apuntan a las imágenes en una vista diferente, y el modo de predicción entre vistas correspondiente se denomina predicción compensada por disparidad (DCP).

En HEVC convencional, hay dos modos para la predicción de los parámetros de movimiento: uno es el modo de fusión y el otro es la predicción avanzada del vector de movimiento (AMVP). En el modo de fusión, se construye una lista de candidatos de parámetros de movimiento (imágenes de referencia y vectores de movimiento) donde el candidato puede ser de bloques contiguos espaciales o temporales. Los bloques contiguos espacialmente y temporalmente pueden formar una lista de candidatos, es decir, un conjunto de candidatos a partir de los cuales se puede seleccionar información de predicción de movimiento. Por consiguiente, el codificador de vídeo 20 puede codificar los parámetros de movimiento elegidos como información de predicción de movimiento codificando un índice en la lista de candidatos. Después de que el descodificador de vídeo 30 haya descodificado el índice, todos los parámetros de movimiento del bloque correspondiente al que el índice apunta pueden ser heredados, en el modo de combinación.

En AMVP, de acuerdo con HEVC convencional, se obtiene una lista de candidatos de predictores de vector de movimiento para cada hipótesis de movimiento basándose en el índice de referencia codificado. Esta lista incluye vectores de movimiento de bloques contiguos que están asociados con el mismo índice de referencia, así como un predictor de vector de movimiento temporal que se obtiene basándose en los parámetros de movimiento del bloque contiguo del bloque colocalizado en una imagen de referencia temporal. Los vectores de movimiento elegidos se señalan transmitiendo un índice a la lista de candidatos. Además, también se señalan los valores del índice de referencia y las diferencias del vector de movimiento.

La FIG. 4 proporciona varios ejemplos de la predicción entre vistas. Las tramas de la vista S1, en el ejemplo de la FIG. 4, se ilustran como predichas a partir de imágenes en diferentes ubicaciones temporales de la vista S1, así como predichas entre vistas a partir de tramas de las tramas de vistas S0 y S2 en las mismas ubicaciones temporales. Por ejemplo, la trama B de la vista S1 en la ubicación temporal T1 se predice a partir de cada una de las tramas B de la vista S1 en las ubicaciones temporales T0 y T2, así como las tramas B de las vistas S0 y S2 en la ubicación temporal T1.

En el ejemplo de la FIG. 4, la letra "B" mayúscula y la "b" minúscula están concebidas para indicar diferentes relaciones jerárquicas entre las tramas, en lugar de diferentes metodologías de codificación. En general, las tramas con "B" mayúscula son relativamente más altas en la jerarquía de predicción que las tramas con "b" minúscula. La FIG. 4 también ilustra las variaciones en la jerarquía de predicción utilizando diferentes niveles de sombreado, donde las tramas con una mayor magnitud de sombreado (es decir, relativamente más oscuras) están más altas en la jerarquía de predicción que aquellas tramas que tienen menos sombreado (es decir, relativamente más claras). Por ejemplo, todas las tramas I en la FIG. 4 se ilustran con sombreado completo, mientras que las tramas P tienen un sombreado algo más claro, y las tramas B (y las tramas con b minúscula) tienen diversos niveles de sombreado, pero siempre más claros que el sombreado de las tramas P y las tramas I.

En general, la jerarquía de predicción se relaciona con índices del orden de vista, en cuanto a que las tramas relativamente más altas en la jerarquía de predicción deberían ser descodificadas antes de la descodificación de tramas que están relativamente más bajas en la jerarquía, de tal modo que esas tramas relativamente más altas en la jerarquía se pueden utilizar como tramas de referencia durante la descodificación de las tramas relativamente más bajas en la jerarquía. Un índice de orden de vista es un índice que indica el orden de descodificación de componentes de vista en una unidad de acceso. Los índices de orden de vista están implícitos en la ampliación SPS MVC, como se especifica en el anexo H de H.264/AVC (la enmienda MVC). En el SPS, para cada índice *i*, se señala el correspondiente *view_id*. En algunos ejemplos, la descodificación de los componentes de vista seguirá el orden ascendente del índice de orden de vista. Si se presentan todas las vistas, los índices de orden de vista se encuentran en un orden consecutivo de 0 a *num_views_minus_1*.

De esta manera, las tramas utilizadas como tramas de referencia pueden ser descodificadas antes de la descodificación de las tramas que se codifican con referencia a las tramas de referencia. Un índice de orden de vista es un índice que indica el orden de descodificación de componentes de vista en una unidad de acceso. Para cada índice de orden de visualización *i*, se señala el *view_id* correspondiente. La descodificación de los componentes de vista sigue el orden ascendente de los índices de orden de vista. Si se presentan todas las vistas, entonces el

conjunto de índices de orden de vista puede comprender un conjunto consecutivamente ordenado de cero a uno menos que el número total de vistas.

5 Para ciertas tramas a niveles iguales de la jerarquía, el orden de descodificación entre ellas puede no importar. Por ejemplo, la trama I de la vista S0 en la posición temporal T0 se utiliza como trama de referencia para la trama P de la vista S2 en la posición temporal T0, que a su vez se utiliza como trama de referencia para la trama P de la vista S4 en la posición temporal T0. En consecuencia, la trama I de la vista S0 en la posición temporal T0 debería descodificarse antes de la trama P de la vista S2 en la posición temporal T0, la cual debería descodificarse antes de la trama P de la vista S4 en la posición temporal T0. Sin embargo, entre las vistas S1 y S3, no importa un orden de descodificación, porque las vistas S1 y S3 no se basan la una en la otra para la predicción, sino que se predicen solo a partir de vistas que son más altas en la jerarquía de predicción. Además, la vista S1 puede descodificarse antes que la vista S4, siempre que la vista S1 sea descodificada después de las vistas S0 y S2.

15 De esta manera, puede utilizarse un ordenamiento jerárquico para describir las vistas S0 a S7. Deje que la notación SA> SB signifique que la vista SA debe ser descodificada antes que la vista SB. Utilizando esta notación, S0> S2> S4> S6> S7, en el ejemplo de la FIG. 4. También con respecto al ejemplo de la FIG. 4, S0> S1, S2> S1, S2> S3, S4> S3, S4> S5 y S6> S5. Cualquier orden de descodificación para las vistas que no viole estos requisitos es posible. En consecuencia, son posibles muchos órdenes de descodificación diferentes con solo ciertas limitaciones.

20 De acuerdo con las técnicas de esta divulgación, cada una de las vistas S0-S7 puede considerarse una capa respectiva de un flujo de bits correspondiente. De este modo, un VPS puede describir parámetros del flujo de bits aplicables a cualquiera de, o todas, las vistas S0-S7, mientras que puedan proporcionarse conjuntos de parámetros de capas individuales para cualquiera de, o todas, las vistas S0-S7. Además, puede proporcionarse un conjunto de parámetros de agrupación para un grupo de conjuntos de parámetros, de manera que los fragmentos dentro de las imágenes individuales de las vistas S0-S7 pueden referirse simplemente al identificador de un conjunto de parámetros de agrupación.

30 Como se muestra en la FIG. 4, un componente de vista puede utilizar los componentes de vista en otras vistas como referencia. Esto se denomina predicción entre vistas. En MVC, la predicción entre vistas se realiza como si el componente de vista en otra vista fuera una referencia de interpredicción. El codificador de vídeo 20 y el descodificador de vídeo 30 pueden codificar las posibles referencias entre vistas en la ampliación MVC del Conjunto de Parámetros de Secuencia (SPS) (como se muestra en el ejemplo de la Tabla 1). El codificador de vídeo 20 y el descodificador de vídeo 30 pueden modificar adicionalmente las posibles referencias entre vistas ejecutando el proceso de construcción de lista de imágenes de referencia, lo cual puede permitir el ordenamiento flexible de las referencias de predicción entre vistas o interpredicción.

TABLA 1

seq_parameter_set_mvc_extension() {	C	Descriptor
num_views_minus1	0	ue(v)
para(i = 0; i <= num_views_minus1; i++)		
view_id[i]	0	ue(v)
para(i = 1; i <= num_views_minus1; i++) {		
num_anchor_refs_l0[i]	0	ue(v)
para(j = 0; j < num_anchor_refs_l0[i]; j++)		
anchor_ref_l0[i][j]	0	ue(v)
num_anchor_refs_l1[i]	0	ue(v)
para(j = 0; j < num_anchor_refs_l1[i]; j++)		
anchor_ref_l1[i][j]	0	ue(v)
}		
para(i = 1; i <= num_views_minus1; i++) {		
num_non_anchor_refs_l0[i]	0	ue(v)
para(j = 0; j < num_non_anchor_refs_l0[i]; j++)		
non_anchor_ref_l0[i][j]	0	ue(v)
num_non_anchor_refs_l1[i]	0	ue(v)
para(j = 0; j < num_non_anchor_refs_l1[i]; j++)		
non_anchor_ref_l1[i][j]	0	ue(v)

}		
num_level_values_signaled_minus1	0	ue(v)
para(i = 0; i <= num_level_values_signaled_minus1; i++) {		
level_idc[i]	0	u(8)
num_applicable_ops_minus1[i]	0	ue(v)
para(j = 0; j <= num_applicable_ops_minus1[i]; j++) {		
applicable_op_temporal_id[i][j]	0	u(3)
applicable_op_num_target_views_minus1[i][j]	0	ue(v)
para(k = 0; k <= applicable_op_num_target_views_minus1[i][j]; k++)		
applicable_op_target_view_id[i][j][k]	0	ue(v)
applicable_op_num_views_minus1[i][j]	0	ue(v)
}		
}		
}		

En la ampliación SPS MVC mostrada en la Tabla 1, para cada vista, se señala el número de vistas que se pueden utilizar para formar la lista de imágenes de referencia 0 y la lista de imágenes de referencia 1. La relación de predicción para una imagen de anclaje, tal como se señala en la ampliación SPS MVC, puede ser diferente de la relación de predicción para una imagen sin anclaje (señalada en la ampliación SPS MVC) de la misma vista.

Entre las normas de codificación de vídeo se incluyen ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 o ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual e ITU-T H.264 (también conocida como ISO/IEC MPEG-4 AVC), incluyendo sus ampliaciones de codificación de vídeo ajustable a escala (SVC) y de codificación de vídeo multivista (MVC).

Además, existe una nueva norma de codificación de vídeo, concretamente la Codificación de Vídeo de Alta Eficiencia (HEVC) que está siendo desarrollada por el Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) del Grupo de Expertos en Codificación de Vídeo (VCEG) de ITU-T y el Grupo de Expertos en Imágenes en Movimiento (MPEG) de ISO/IEC. Un reciente Borrador de Trabajo (BT) de la HEVC, y denominado HEVC WD4 en adelante, está disponible en http://phenix.int-evry.fr/jct/doc_end_user/documents/6_Torino/wg11/JCTVC-F803-v3.zip, denotado como HEVC WD4d1.

El mecanismo de conjunto de parámetros de secuencia e imagen desacopla la transmisión de información que cambia poco frecuentemente de la transmisión de datos de bloque codificados. En algunas aplicaciones, los conjuntos de parámetros de secuencia e imagen pueden ser transportados "fuera de banda" usando un mecanismo de transporte fiable. Una carga útil de secuencia de bytes sin procesar (RBSP) de un conjunto de parámetros de imagen puede incluir parámetros a los que pueden referirse las unidades de capa de abstracción de red de fragmento codificado (NAL) de una o más imágenes codificadas. Una o más RBSP de un conjunto de parámetros de secuencia puede incluir parámetros a los que puede referirse una o más RBSP de un conjunto de parámetros de imagen o una o más unidades NAL de información de mejora complementaria (SEI) que contienen un mensaje SEI de periodo de almacenamiento intermedio. Una RBSP de un conjunto de parámetros de secuencia puede incluir parámetros a los que puede hacer referencia una o más RBSP de un conjunto de parámetros de imagen o una o más unidades SEI NAL que contienen un mensaje SEI de periodo de almacenamiento intermedio.

El conjunto de parámetros de secuencia puede incluir un conjunto opcional de parámetros denominado información de usabilidad de vídeo (VUI). La VUI puede incluir las siguientes tres categorías de información opcional: información de representación de vídeo, información del decodificador de referencia hipotética (HRD) e información de restricción del flujo de bits. La información de representación de vídeo incluye la relación de aspecto, cambios de fase de croma de información de transformación de espacio de color relativos a luma y velocidad de tramas. El HRD incluye parámetros de almacenamiento intermedio de vídeo para las secuencias de vídeo codificadas. La información de restricción del flujo de bits incluye restricciones sobre el rango del vector de movimiento, el tamaño de la memoria intermedia de imágenes descodificadas (DPB) y el número de tramas de reordenación y los tamaños codificados de bloques (por ejemplo, macrobloques o unidades de codificación (CU)) e imágenes.

HEVC WD5 incluye el conjunto de parámetros de adaptación de soporte (APS). El concepto de conjunto de parámetros de adaptación (APS) también se puede encontrar en JCTVC-F747, disponible en http://phenix.int-evry.fr/jct/doc_end_user/documents/6_Torino/wg11/JCTVC-F747-v4.zip.

Se puede utilizar una cabecera de unidad NAL unificada tanto para los flujos de bits HEVC no escalables como para

los flujos de bits escalables que se ajustan a las ampliaciones potenciales escalables o multivista de HEVC. Una cabecera de unidad NAL unificada puede diferir de la cabecera de unidad HEVC NAL actual en los siguientes aspectos: puede haber una longitud de cabecera de unidad NAL fija para una secuencia de vídeo codificada completa, mientras que la longitud puede variar entre secuencias de vídeo codificadas diferentes, y una codificación eficaz de los elementos sintácticos de escalabilidad en la cabecera de unidad NAL y, cuando un elemento sintáctico particular no es necesario, no necesita estar presente. En tal diseño, se puede usar un tipo de unidad NAL o un conjunto de parámetros diferente para todo el flujo de bits.

La FIG. 5 es un diagrama conceptual que ilustra un conjunto de parámetros de vídeo (VPS) y varios conjuntos de parámetros de capa (LPS). Las elipses que siguen al segundo LPS de la FIG. 5 pretenden indicar que puede haber cualquier número N de VPS, donde N es un número entero. Por ejemplo, cada capa (por ejemplo, cada capa SVC o cada vista MVC) puede tener un LPS correspondiente. Un codificador de vídeo, tal como el codificador de vídeo 20 o el descodificador de vídeo 30, puede configurarse para codificar un VPS y uno o más LPS, como los ilustrados en la FIG. 5.

La Tabla 2 siguiente proporciona un ejemplo de sintaxis de carga útil de secuencia de bytes sin procesar (RBSP) para un VPS.

TABLA 2

video_parameter_set_rbsp() {	Descriptor
video_para_set_id	u(8)
// descripción del contador de dimensión de la muestra	
cnt_p	u(3)
cnt_d	u(3)
cnt_t	u(3)
cnt_q	u(3)
cnt_v	u(4)
cnt_f	u(4)
// índice de muestra para la asignación de características	
para(i = 0; i < cnt_d; i++) {	
pic_width_in_luma_samples[i]	ue(v)
pic_height_in_luma_samples[i]	ue(v)
bit_depth_luma_minus8[i]	ue(v)
bit_depth_chroma_minus8[i]	ue(v)
chroma_format_idc[i]	u(2)
}	
para(i = 0; i < cnt_t; i++)	
average_frame_rate[i]	u(16)
si(cnt_v > 1)	
para(i=0; i < cnt_v; i++)	
view_id[i]	ue(v)
// parámetros de control e indicadores de habilitación / deshabilitación de herramientas	
log2_max_pic_order_cnt_lsb_minus4	ue(v)
chroma_pred_from_luma_enabled_flag	u(1)
loop_filter_across_slice_flag	u(1)
sample_adaptive_offset_enabled_flag	u(1)
adaptive_loop_filter_enabled_flag	u(1)
pcm_loop_filter_disable_flag	u(1)
cu_qp_delta_enabled_flag	u(1)

temporal_id_nesting_flag	u(l)
inter_4x4_enabled_flag	u(l)
operation_point_description()	
vps_extension_flag	u(l)
si(vps_extension_flag)	
mientras(more_rbsp_data())	
vps_extension_data_flag	u(l)
rbbsp_trailing_bits()	
}	

Los codificadores de vídeo se pueden configurar de tal manera que una secuencia de vídeo codificada (por ejemplo, un flujo de bits que incluya una o más capas) solo pueda tener un conjunto de parámetros de vídeo (VPS) activo. El VPS puede estar encapsulado dentro de una unidad NAL de un tipo particular. Por ejemplo, el nal_unit_type para un VPS RBSP puede ser 10. La semántica de ejemplo para el VPS de la Tabla 2 se describe a continuación:

En este ejemplo, video_para_set_id identifica un conjunto de parámetros de vídeo (VPS) correspondiente.

En este ejemplo, cnt_p especifica el número máximo de valores de priority_id presentes en la secuencia de vídeo codificada correspondiente.

En este ejemplo, cnt_d especifica el número máximo de capas de dependencia presentes en la secuencia de vídeo codificada correspondiente. Varias vistas con la misma resolución pueden considerarse como pertenecientes a una misma capa de dependencia. Dos capas de dependencia pueden tener la misma resolución espacial.

En este ejemplo, cnt_t especifica el número máximo de capas temporales presentes en la secuencia de vídeo codificada.

En este ejemplo, cnt_q especifica el número máximo de capas de calidad presentes en una capa de dependencia en la secuencia de vídeo codificada.

En este ejemplo, cnt_v especifica el número máximo de vistas presentes en la secuencia de vídeo codificada.

En este ejemplo, cnt_f especifica el número de bits utilizados para representar el elemento sintaxis reserved_flags en la cabecera de la unidad NAL.

En este ejemplo, pic_width_in_luma_samples [i] y pic_height_in_luma_samples [i] especifican, respectivamente, la anchura y la altura de la i-ésima resolución de la capa de dependencia en unidades de muestras de luma.

En este ejemplo, bit_depth_luma_minus8 [i] más 8 y bit_depth_chroma_minus8 [i] más 8 especifica la profundidad de bits de los componentes luma y croma de la i-ésima representación de profundidad de bits.

En este ejemplo, chroma_format_idc [i] especifica el formato de la muestra de croma de la i-ésima representación del formato de la muestra de croma. Por ejemplo, un valor igual a 0 puede indicar 4:2:0; un valor igual a 1 puede indicar 4:4:4, un valor igual a 2 puede indicar 4:2:2 y un valor igual a 3 puede indicar 4:0:0.

En este ejemplo, average_frame_rate[i] especifica la velocidad media de la i-ésima representación de la capa temporal en unidades de tramas por 256 segundos.

En este ejemplo, view_id[i] especifica el identificador de vista de la i-ésima vista, que tiene índice de orden de vista igual a i. Cuando no está presente, se puede inferir que el valor de view_id[0] es 0. vps_extension_flag igual a 0 especifica que no hay elementos sintácticos vps_extension_data_flag presentes en la estructura de sintaxis RBSP del conjunto de parámetros de vídeo. vps_extension_flag puede ser igual a 0 en flujos de bits conforme a la próxima norma HEVC. El valor 1 para vps_extension_flag puede reservarse, por ejemplo, para un uso futuro mediante ITU-T | ISO/IEC. Los descodificadores, como el descodificador de vídeo 30, pueden ignorar todos los datos que siguen al valor 1 para vps_extension_flag en una unidad NAL del conjunto de parámetros de vídeo.

En este ejemplo, vps_extension_data_flag puede tener cualquier valor. No afecta a la conformidad con los perfiles especificados en la próxima norma HEVC, pero permite el desarrollo posterior de la próxima norma.

Otros elementos sintácticos en el VPS pueden tener la misma semántica que los elementos sintácticos con los mismos nombres en el SPS del borrador de trabajo HEVC actual. Esos elementos sintácticos pueden aplicarse a la secuencia de vídeo codificada que hace referencia a este VPS, a menos que sean sobrescritos por conjuntos de

parámetros de nivel inferior.

En algunos ejemplos, una señal 3DV_flag puede indicarse adicionalmente en el VPS para indicar si la profundidad está presente en la secuencia de vídeo codificada.

5 En algunos ejemplos, los parámetros VUI se señalan en el LPS.

10 En algunos ejemplos, los elementos sintácticos cnt_p, cnt_t, cnt_d, cnt_q y cnt_v especifican los números de bits utilizados para codificar priority_id, temporal_id, dependency_id, quality_id y view_idx, respectivamente, y los números máximos de valores priority_id, capas temporales, capas de dependencia, capas de calidad y vistas presentes en las secuencias de vídeo codificadas también se pueden señalar en el VPS.

15 En algunos ejemplos, se puede introducir otro tipo de unidad NAL para contener los elementos sintácticos cnt_p, cnt_t, cnt_d, cnt_q, cnt_v y cnt_f. Este nuevo tipo de unidad NAL también puede incluir un identificador (ID) y puede hacerse referencia al ID en el VPS.

20 En algunos ejemplos, los elementos sintácticos de log2_max_pic_order_cnt_lsb_minus4 a inter_4x4_enabled_flag en la Tabla 2 no están señalados en el VPS, sino que en su lugar, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden codificar estos elementos sintácticos en el LPS.

25 En algunos ejemplos, la estructura de sintaxis operation_point_description () de la Tabla 2 no está incluida en el VPS; en su lugar, el codificador de vídeo 20 y el descodificador de vídeo 30 u otros elementos (por ejemplo, la interfaz de salida 22 y/o la interfaz de entrada 28), pueden codificar el contenido en la estructura de sintaxis de operation_point_description () en un mensaje de información de mejora complementaria (SEI).

30 En algunos ejemplos, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden codificar los parámetros de información de usabilidad de vídeo (VUI) en el VPS. Por ejemplo, un VPS puede incluir datos que especifican información de restricción del flujo de bits, tales como restricciones en el rango de vectores de movimiento, tamaño DPB, número de tramas de reordenamiento y tamaños codificados de bloques (por ejemplo, macrobloques o CU) e imágenes. De esta manera, un VPS puede especificar información que indica un tamaño DPB requerido para que un descodificador de vídeo (tal como el descodificador de vídeo 30) descodifique adecuadamente un flujo de bits correspondiente, es decir, un flujo de bits que incluya el VPS. Del mismo modo, un VPS puede especificar información de reordenación de imagen, es decir, un cierto número de imágenes que pueden preceder a una imagen dada en orden de descodificación y que suceden a la imagen dada en orden de salida (es decir, orden de visualización).

40 Adicionalmente o de forma alternativa, un VPS puede incluir datos que especifican información de descodificador de referencia hipotética (HRD). Como se ha indicado anteriormente, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden codificar (es decir, señalar) los parámetros VUI, que pueden incluir información HRD en el VPS. Por lo tanto, un VPS puede incluir datos que describen, por ejemplo, puntos de operación de un flujo de bits correspondiente. Por ejemplo, un VPS puede incluir datos que describen uno o más de un cierto número de puntos de operación máximos, dependencias entre capas o vistas diferentes, información de perfil y nivel para cada punto de operación, representación de unidad de VCL NAL de punto de operación para cada punto de operación, velocidad de bits para cada punto de operación, dependencia entre puntos de operación, restricciones para cada punto de operación, VUI o VUI parcial para cada punto de operación y/o VUI o VUI parcial para cada capa o vista.

50 Un VPS también puede incluir para cada dimensión: un valor de índice específico, un intervalo de valores de índice o una lista de valores de índice. Por ejemplo, cuando un VPS incluye datos que describen un valor de índice específico, el valor de índice puede corresponder, para una resolución espacial, a la profundidad de bits para el formato de muestreo de croma. Para mencionar otro ejemplo, cuando un VPS incluye un intervalo de valores de índice, para capas temporales, el intervalo puede comprender desde cero (0) al ID de capa temporal más alto, y para capas de calidad, el intervalo puede comprender desde cero (0) al ID de capa de calidad más alto. Como otro ejemplo, cuando un VPS incluye datos que describen una lista de valores de índice, la lista puede comprender una lista de valores de índice de vista para múltiples vistas.

60 En algunos ejemplos, el codificador de vídeo 20 puede codificar (es decir, señalar) y el descodificador de vídeo puede descodificar, uno o más parámetros de formato de representación (anchura, altura, profundidad de bits, etc.) y puede haber diferentes conjuntos de parámetros de formato de representación. Una capa o punto de operación puede entonces referirse a un índice de dicho conjunto de parámetros de formato de representación. Un ejemplo del diseño de sintaxis para tal conjunto se muestra en la Tabla 3 siguiente.

TABLA 3

num_rep_formats_minus1	ue(v)
------------------------	-------

para(i = 0; i <= num_rep_formats_minus1; i++) {	
pic_width_in_luma_samples[i]	ue(v)
pic_height_in_luma_samples[i]	ue(v)
bit_depth_luma_minus8[i]	ue(v)
bit_depth_chroma_minus8[i]	ue(v)
chroma_format_idc[i]	u(2)
}	
para(i = 0; i < cnt_d; i++) {	
rep_format_idx[i]	ue(v)
}	

En algunos ejemplos, en lugar de eso, el ref_format_idx puede señalarse en el conjunto de parámetros de capa.

La Tabla 4 siguiente proporciona una sintaxis de ejemplo para las descripciones de punto de operación.

5

TABLA 4

operation_points_description() {	Descriptor
num_operation_point_minus1	ue(v)
para(i = 0; i <= num_operation_points_minus1; i++) {	
op_profile_level_idc[i]	u(24)
operation_point_id[i]	ue(v)
priority_id[i]	ue(v)
temporal_id[i]	ue(v)
quality_id[i]	ue(v)
dependency_id[i]	ue(v)
if (cnt_v > 1) {	
num_target_output_views_minus1[i]	ue(v)
para(j = 0; j <= num_target_output_views_minus1[i]; j++)	
view_idx[i][j]	ue(v)
}	
frm_rate_info_present_flag[i]	u(1)
avg_bitrate[i]	u(16)
max_bitrate[i]	u(16)
max_bitrate_calc_window[i]	u(16)
constant_frm_rate_idc[i]	u(2)
si (cnt_v > 1) {	
num_directly_dependent_views[i]	
para(j = 0; j < num_directly_dependent_views[i]; j++) {	
directly_dependent_view_idx[i][j]	
}	
}	
si (cnt_v > 1)	
para(i = 1; i < cnt_v; i++) {	
num_ref_views[i]	ue(v)
para(j = 0; j < num_ref_views[i]; j++)	
ref_view_idx[i][j]	ue(v)

}	
}	

A continuación se analizan ejemplos de la semántica de los elementos sintácticos de la Tabla 4:

5 En este ejemplo, num_operation_point_minus1 más 1 especifica el número máximo de puntos de operación que están presentes en la secuencia de vídeo codificada y para los cuales la información de punto de operación es señalada por los siguientes elementos sintácticos.

10 En este ejemplo, op_profile_level_idc [i], operation_point_id [i], priority_id [i], num_target_output_views_minus1 [i], frm_rate_info_present_flag [i], avg_bitrate [i], max_bitrate [i], max_bitrate_calc_window [i], constant_frm_rate_idc [i] y num_directly_dependent_views [i] puede tener la misma semántica que los elementos sintácticos con los mismos nombres en el mensaje SEI de información de escalabilidad de H.264.

15 En este ejemplo, quality_id [i] y dependency_id [i] pueden tener la misma semántica que los elementos sintácticos con los mismos nombres en el mensaje SEI de información de escalabilidad de H.264.

En este ejemplo, directly_dependent_view_idx[i][j] especifica el índice de vista de la j-ésima vista de la cual depende directamente la vista de salida de destino del punto de operación actual en la representación del punto de operación actual.

20 En este ejemplo, num_ref_views[i] especifica el número de componentes de vista para la predicción entre vistas en la lista de imágenes de referencia inicial RefPicList0 y RefPicList1 en los componentes de vista de descodificación con índice de orden de vista igual a i. En este ejemplo, el valor de las vistas num_ref[i] no será mayor que Min (15, num_views_minus1). En algunos ejemplos, el valor de num_ref_views[0] es igual a 0.

25 En este ejemplo, ref_view_idx[i][j] especifica el índice de orden de vista del j-ésimo componente de vista para la predicción entre vistas en la lista de imágenes de referencia inicial RefPicList0 y RefPicList1 en la descodificación de un componente de vista con índice de orden de vista igual a i. En este ejemplo, el valor de ref_view_idx[i][j] estará en el rango de 0 a 31, inclusive.

30 En algunos ejemplos, como alternativa, algunos de los elementos sintácticos del mensaje SEI de información de escalabilidad (por ejemplo, como se describe en H.264), por ejemplo, los elementos sintácticos relacionados con la información de dependencia de capa, se pueden incluir en la estructura de sintaxis operation_points_description() de la Tabla 4.

35 En algunos ejemplos, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden codificar (es decir, señalar) algunos parámetros VUI en la estructura de sintaxis operation_points_description() de la Tabla 4.

La Tabla 5 siguiente proporciona una sintaxis alternativa para un conjunto de parámetros de vídeo:

40

TABLA 5

video_parameter_set_rbsp() {	Descriptor
video_parameter_set_id	ue(v)
num_temporal_layers_minus1	u(3)
para (i = 0; i <= num_temporal_layers_minus1; i++) {	
profile_idc[i]	u(8)
reserved_zero_8bits[i] /* equal to 0 */	u(8)
level_idc[i]	u(8)
}	
bit_depth_luma_minus8	ue(v)
bit_depth_chroma_minus8	ue(v)
chroma_format_idc	u(2)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
pic_cropping_flag	u(1)
si(pic_cropping_flag) {	

pic_crop_left_offset	ue(v)
pic_crop_right_offset	ue(v)
pic_crop_top_offset	ue(v)
pic_crop_bottom_offset	ue(v)
}	
temporal_id_nesting_flag	u(1)
bit_equal_to_one /* igual a 1 */	u(1)
extension_type /* igual a 0 para 3DV */	ue(v)
num_layers_minus2	ue(v)
num_rep_formats_minus1	ue(v)
para(i = 1; i <= num_rep_formats_minus1; i++) {	
bit_depth_luma_minus8[i]	ue(v)
bit_depth_chroma_minus8[i]	ue(v)
chroma_format_idc[i]	u(2)
pic_width_in_luma_samples[i]	ue(v)
pic_height_in_luma_samples[i]	ue(v)
pic_cropping_flag[i]	u(1)
si(pic_cropping_flag[i]) {	
pic_crop_left_offset[i]	ue(v)
pic_crop_right_offset[i]	ue(v)
pic_crop_top_offset[i]	ue(v)
pic_crop_bottom_offset[i]	ue(v)
}	
}	
para(i = 1; i <= num_layers_minus1; i++) {	
rep_format_idx[i]	ue(v)
si(extension_type == 1) {	
dependency_id[i]	ue(v)
quality_id[i]	ue(v)
num_directly_dependent_layers[i]	ue(v)
para(j = 0; j < num_directly_dependency_layers[i]; j++)	
delta_reference_layer_id_minus1[i][j]	ue(v)
}	
}	
num_short_term_ref_pic_sets	ue(v)
para(i = 0; i < num_short_term_ref_pic_sets; i++)	
short_term_ref_pic_set(i)	
si (extension_type == 0)	
view_dependency()	
num_additional_profiles_levels_minus1	ue(v)
para(i = 0; i <= num_additional_profiles_levels_minus1; i++) {	
additional_profile_idc[i]	u(8)
additional_reserved_zero_8bits[i] /* igual a 0 */	u(8)
additional_level_idc[i]	u(8)
num_applicable_operation_points_minus1[i]	ue(v)
para(j = 0; j <= num_applicable_operation_points[i]; j++) {	

temporal_id[i][j]	ue(v)
layer_id[i][j]	ue(v)
si(extension_type == 0) { /* Siempre verdadero para 3DV */	
depth_included_flag	u(1)
num_target_output_views_minus1[i][j]	ue(v)
para(k = 0; k < num_target_output_views_minus1[i][j]; k++)	
layer_id[i][j][k]	ue(v)
}	
si no (extension_type == 1)	
layer_id[i][j]	ue(v)
}	
}	
vps_vui_parameters_present_flag	u(1)
si(vps_vui_parameters_present_flag)	
vps_vui_parameters()	
vps_extension_flag	u(1)
si(vps_extension_flag)	
mientras(more_rbsp_data())	
vps_extension_data_flag	u(1)
rbps_trailing_bits()	
}	

Ejemplos de la semántica para la sintaxis del conjunto de parámetros de vídeo de la Tabla 5 se analizan continuación. En general, los elementos sintácticos nombrados de forma similar que no se analizan a continuación pueden tener la misma semántica como se ha analizado anteriormente con respecto a la Tabla 2. La semántica para los otros elementos sintácticos puede ser la siguiente:

En este ejemplo, bit_equal_to_one es igual a 1 (es decir, un valor binario "1").

En este ejemplo, extension_type igual a 0 indica que puede haber capas multivista en el flujo de bits. En este ejemplo, extension_type igual a 1 especifica que varias capas de dependencia y/o calidad pueden estar presentes en el flujo de bits.

En este ejemplo, num_rep_formats_minus1 más 1 especifica el número máximo de formatos de representación de conjuntos diferentes compatibles con este conjunto de parámetros de vídeo; un formato de representación incluye la profundidad de bits y el formato de croma (es decir, los conjuntos de valores de bit_depth_luma_minus8, bit_depth_chroma_minus8 y chroma_format_idc), información sobre la resolución de imagen y la ventana de recorte en la secuencia de vídeo codificada. El valor del num_rep_formats_minus1 puede estar en la gama entre 0 y X, inclusive. El codificador de vídeo 20 y el decodificador de vídeo 30 pueden codificar el conjunto de profundidad de bits y el formato de croma para la capa base mediante bit_depth_luma_minus8, bit_depth_chroma_minus8 y chroma_format_idc y los conjuntos de profundidad de bits y formato de croma son señalados para capas de realce mediante el siguiente conjunto de elementos sintácticos bit_depth_luma_minus8[i], bit_depth_chroma_minus8[i] y chroma_format_idc[i].

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden codificar el primer conjunto de formato de representación mediante muestras bit_depth_luma_minus8, bit_depth_chroma_minus8, chroma_format_idc, pic_width_in_luma, pic_height_in_luma_samples, pic_cropping_flag, pic_crop_left_offset, pic_crop_right_offset, pic_crop_top_offset y pic_crop_bottom_offset.

En este ejemplo, bit_depth_luma_minus8[i], bit_depth_chroma_minus8[i] y chroma_format_idc[i] especifican, respectivamente, el i-ésimo conjunto de valores bit_depth_luma_minus8, bit_depth_chroma_minus8 y chroma_format_idc en la secuencia de vídeo codificada.

En este ejemplo, pic_width_in_luma_samples[i] y pic_height_in_luma_samples[i] especifican, respectivamente, la anchura y la altura de cada imagen decodificada en unidades de muestras de luma utilizando el i-ésimo formato de representación.

En este ejemplo, `pic_cropping_flag [i]`, `pic_crop_left_offset [i]`, `pic_crop_right_offset [i]`, `pic_crop_top_offset [i]` y `pic_crop_bottom_offset [i]` especifican, para el *i*-ésimo conjunto de formato de representación, las muestras de las imágenes en la secuencia de vídeo codificada que se emiten desde el proceso de descodificación, en términos de una región rectangular especificada en coordenadas de imagen para la salida.

En este ejemplo, `rep_format_idx [i]` especifica el índice de valores para el conjunto de profundidad de bits adicional y el formato de croma que se aplica a la capa con `layer_id` igual a *i*. Los valores de `bit_depth_luma_minus8`, `bit_depth_chroma_minus8` y `chroma_format_idc` para la capa con `layer_id` igual a *i* pueden ser iguales a `bit_depth_luma_minus8 [rep_format_idx[i]]`, `bit_depth_chroma_minus8 [rep_format_idx[i]]` y `chroma_format_idc [rep_format_idx[i]]`, respectivamente. El valor de `rep_format_idx[i]` estará en la gama entre 0 y *X*, inclusive.

En este ejemplo, `dependency_id[i]` especifica un identificador de dependencia para la capa con `layer_id` igual a *i*. `dependency_id[i]` puede estar en el rango de 0 a *X* inclusive. Cuando no está presente, se puede inferir que `dependency_id[i]` que es 0. Cuando `num_directly_dependent_layers[i]` es mayor que 0, `dependency_id[i]` puede ser igual o mayor que el identificador de dependencia de cualquier capa de la cual dependa la capa con `layer_id` igual a *i*.

En este ejemplo, `quality_id[i]` especifica un identificador de igualdad para la capa con `layer_id` igual a *i*. `quality_id[i]` puede estar en el rango de 0 a *X* inclusive. Cuando no está presente, se puede inferir que `quality_id [i]` es 0. Cuando `num_directly_dependent_layers[i]` es mayor que 0, `quality_id[i]` puede ser igual o mayor que el identificador de calidad de cualquier capa de la cual dependa la capa con `layer_id` igual a *i* y que tenga un identificador de dependencia igual a `dependency_id[i]`.

En este ejemplo, `num_short_term_ref_pic_sets` especifica el número de conjuntos de imágenes de referencia a corto plazo que se especifican en el conjunto de parámetros de vídeo. El valor de `num_short_term_ref_pic_sets` puede estar en la gama entre 0 y 64, inclusive.

En este ejemplo, `depth_included_flag` igual a 1 indica que el punto de operación 3DV actual contiene profundidad. En este ejemplo, `depth_included_flag` igual a 0 indica que el punto de operación 3DV actual no contiene profundidad.

La sintaxis de ejemplo para el elemento de dependencia de vista de la Tabla 5 se proporciona en la Tabla 6 siguiente:

TABLA 6

<code>view_dependency() {</code>	
<code> num_views_minus1</code>	<code>ue(v)</code>
<code> para(i = 0; i <= num_views_minus1; i++)</code>	
<code> view_id[i]</code>	<code>ue(v)</code>
<code> para(i = 1; i <= num_views_minus1; i++) {</code>	
<code> num_ref_views[i]</code>	
<code> para(j = 0; j < num_ref_views[i]; j++)</code>	
<code> ref_view_idx[i][j]</code>	<code>ue(v)</code>
<code> inter_view_texture_flag[i][j]</code>	<code>u(1)</code>
<code> }</code>	
<code> }</code>	
<code>}</code>	

La Tabla 7 siguiente define un ejemplo de un conjunto de datos en los que la dependencia de vista de cada vista no básica se señala directamente en el nivel de secuencia.

TABLA 7

<code>para(i = 1; i <= num_views_minus1; i++)</code>	
<code>{</code>	
<code> num_ref_views[i]</code>	<code>ue(v)</code>
<code> para(j = 0; j < num_ref_views[i]; j++)</code>	
<code> }</code>	

ref_view_idx[i][j]	ue(v)
}	

En este ejemplo, num_ref_views[i] especifica el número de componentes de vista para la predicción entre vistas en la lista de imágenes de referencia inicial RefPicList0 y RefPicList1 en los componentes de vista de decodificación con índice de orden de vista igual a i. En este ejemplo, el valor de num_ref_views[i] no es mayor que Min(15, num_views_minus1). En este ejemplo, el valor de num_ref_views[0] es igual a 0.

En este ejemplo, ref_view_idx[i][j] especifica el índice de orden de vista del j-ésimo componente de vista para la predicción entre vistas en la lista de imágenes de referencia inicial RefPicList0 y RefPicList1 en la decodificación de un componente de vista con índice de orden de vista igual a i. En este ejemplo, el valor de ref_view_idx[i][j] está en el rango de 0 a 31, inclusive.

Como se ha indicado anteriormente, puede usarse un tipo particular de unidad NAL (por ejemplo, tipo de unidad NAL 10) para encapsular un conjunto de parámetros de vídeo. La sintaxis de la unidad NAL puede modificarse como se muestra en el ejemplo de la Tabla 8 siguiente.

TABLA 8

nal_unit(NumBytesInNALunit) {	Descriptor
forbidden_zero_bit	f(1)
nal_ref_flag	u(1)
nal_unit_type	u(6)
NumBytesInRBSP = 0	
nalUnitHeaderBytes = 1	
si(nal_unit_type != 10) { // no unidad VPS NAL	
si(cnt_p > 1)	
id de prioridad	u(v)
si(cnt_t > 1)	
temporal_id	u(v)
reserved_one_bit	u(1)
si(cnt_d > 1)	
dependency_id	u(v)
si(cnt_q > 1)	
quality_id	u(v)
reserved_one_bit	u(1)
si(cnt_v > 1)	
view_idx	u(v)
si(cnt_f)	
reserved_flags	u(v)
m = Ceil(log2(cnt_p)) + Ceil(log2(cnt_t)) + Ceil(log2(cnt_d)) + Ceil(log2(cnt_q)) + Ceil(log2(cnt_v)) + cnt_f + 2	
si(((m + 7 >> 3) << 3) - m)	
bits reservados	u(v)
nalUnitHeaderBytes += ((m + 7) >> 3)	
}	
para(i = nalUnitHeaderBytes; i < NumBytesInNALunit; i++) {	
si(i + 2 < NumBytesInNALunit && next_bits(24) == 0x000003) {	
rbsp_byte [NumBytesInRBSP++]	b(8)
rbsp_byte [NumBytesInRBSP++]	b(8)
i+=2	

emulation_prevention_three_byte /* igual a 0x03 */	f(8)
si no	
rbsp_byte [NumBytesInRBSP++]	b(8)
}	
}	

En este ejemplo, se añaden elementos dentro de la instrucción "si(nal_unit_type!=10)", en relación con la sintaxis de la unidad NAL convencional. En este ejemplo, el número de bits utilizados para señalar los elementos sintácticos priority_id, temporal_id, dependency_id, quality_id y view_idx es Ceil(log2 (cnt_p)), Ceil(log2 (cnt_t)), Ceil(log2 (cnt_d)), Ceil(log2(cnt_q)) y Ceil(log2(cnt_v)), respectivamente. Del mismo modo, en este ejemplo, cuando cualquiera de los elementos sintácticos priority_id, temporal_id, dependency_id, quality_id y view_idx no está presente, se deduce que el valor de ese elemento sintáctico es igual a 0.

Excepto como se ha definido anteriormente con respecto al número de bits y las inferencias que pueden dibujarse, la semántica para los elementos sintácticos de la Tabla 8 puede definirse de la forma siguiente. La semántica de priority_id, dependency_id y quality_id puede ser tal como se define en la ampliación SVC de ITU-T H.264/AVC. La semántica de temporal_id puede ser como se define en WD4 de HEVC. En este ejemplo, reserved_one_bit es igual a 1. El valor 0 para reserved_one_bit puede ser especificado por futuras ampliaciones de la norma HEVC. Los decodificadores, tales como el decodificador de vídeo 30, pueden configurarse para ignorar el valor de reserved_one_bit.

En este ejemplo, view_idx especifica el índice de orden de vista para una vista. La semántica de view_idx puede ser la misma que el elemento sintáctico "índice de orden de vista" tal como se especifica en la ampliación MVC de ITU-T H.264/AVC.

En este ejemplo, cada bit de reserved_flags es igual a 1. Otros valores para reserved_flags pueden ser especificados por futuras ampliaciones de la próxima norma HEVC. Los decodificadores, tales como el decodificador de vídeo 30, pueden configurarse para ignorar el valor de reserved_flags, a menos que estén configurados para funcionar de acuerdo con una ampliación que asigne semántica a bits de reserved_flags. En este ejemplo, el número de bits utilizados para representar reserved_flags es reserved_flags_len.

En este ejemplo, cada bit de reserved_bits es igual a 1. Otros valores para reserved_bits pueden ser especificados por la ampliación futura de la próxima norma HEVC. Los decodificadores, tales como el decodificador de vídeo 30, pueden configurarse para ignorar el valor de reserved_bits, de nuevo a menos que estén configurados de acuerdo con dicha ampliación futura. El número de bits utilizados para representar reserved_bits, en este ejemplo, es ((m+7 >> 3) << 3) -m.

La Tabla 9 siguiente proporciona una sintaxis de ejemplo para un conjunto de parámetros de capa. Se puede usar la misma sintaxis para cada uno de los LPS de la FIG. 5, en algunos ejemplos.

TABLA 9

layer_para_set() {	Descriptor
depth_flag	u(1)
layer_para_set_id	ue(v)
vps_id	ue(v)
// cu hierarchy {{	
log2_min_coding_block_size_minus3	ue(v)
log2_diff_max_min_coding_block_size	ue(v)
log2_min_transform_block_size_minus2	ue(v)
log2_diff_max_min_transform_block_size	ue(v)
log2_min_pcm_coding_block_size_minus3	ue(v)
max_transform_hierarchy_depth_inter	ue(v)
max_transform_hierarchy_depth_intra	ue(v)
// jerarquía de cu }}	
pcm_bit_depth_luma_minus1	u(4)

pcm_bit_depth_chroma_minus1	u(4)
loop_filter_across_slice_flag	u(1)
sample_adaptive_offset_enabled_flag	u(1)
adaptive_loop_filter_enabled_flag	u(1)
pcm_loop_filter_disable_flag	u(1)
cu_qp_delta_enabled_flag	u(1)
// componentes	
num_tile_columns_minus1	ue(v)
num_tile_rows_minus1	ue(v)
si (num_tile_columns_minus1 != 0 num_tile_rows_minus1 0) {	
tile_boundary_independence_idc	u(1)
uniform_spacing_idc	u(1)
si (uniform_spacing_idc != 1) {	
para (i=0; i<num_tile_columns_minus1 ; i++)	
anchura de columna[i]	ue(v)
para (i=0; i < num_tile_rows_minus1; i++)	
row_height[i]	ue(v)
}	
}	
lps_extension_flag	u(1)
si(lps_extension_flag)	
mientras(more_rbsp_data())	
lps_extension_data_flag	u(1)
rbps_trailing_bits()	
}	

A continuación se describen ejemplos de la semántica para la sintaxis LPS de la Tabla 9. Diferentes capas (por ejemplo, diferentes vistas en MVC o diferentes capas en SVC) pueden referirse a diferentes LPS. Diferentes capas de calidad en una misma capa de dependencia pueden compartir el mismo LPS. Diferentes capas temporales en una misma capa de dependencia pueden compartir el mismo LPS. De forma alternativa, diferentes vistas pueden referirse a un mismo LPS y diferentes capas de dependencia pueden referirse a un mismo LPS.

En este ejemplo, depth_flag igual a 1 especifica que el LPS se aplica a las representaciones de profundidad identificadas por los valores de temporal_id, dependency_id, quality_id y view_idx de la unidad LPS NAL. Depth_flag igual a 0 especifica que el LPS se aplica a las representaciones de textura identificadas por los valores de temporal_id, dependency_id, quality_id y view_idx de la unidad LPS NAL.

En este ejemplo, layer_para_set_id especifica el id del conjunto de parámetros de capa actual (LPS). Diferentes conjuntos de parámetros de capa con los mismos valores de dependency_id y view_idx, respectivamente, comparten un espacio de valor para layer_para_set_id, lo cual significa que diferentes LPS con diferentes combinaciones de depecey_id y view_idx pueden tener el mismo valor de layer_para_set_id.

De forma alternativa, todos los LPS pueden compartir el espacio de un valor, lo cual significa que cada LPS tiene un valor distinto de layer_para_set_id.

En este ejemplo, vps_id identifica el conjunto de parámetros de vídeo al que se refiere este conjunto de parámetros de capa.

En este ejemplo, lps_extension_flag igual a 0 especifica que no hay elementos sintácticos lps_extension_data_flag presentes en la estructura de sintaxis RBSP del conjunto de parámetros de capa. En este ejemplo, lps_extension_flag puede ser igual a 0 en flujos de bits que se ajustan a la próxima norma HEVC. El valor 1 para

lps_extension_flag está reservado para un uso futuro por ITU-T | ISO/IEC. Los decodificadores, como el decodificador de vídeo 30, pueden ignorar todos los datos que siguen al valor 1 para lps_extension_flag en una unidad NAL de conjunto de parámetros de capa.

- 5 En este ejemplo, lps_extension_data_flag puede tener cualquier valor y no afecta la conformidad con los perfiles especificados en la próxima norma HEVC.

Otros elementos sintácticos pueden tener la misma semántica que los elementos sintácticos con los mismos nombres en el SPS de la HEVC WD, pero aplicando solo a las imágenes que se refieren a este LPS.

- 10 Un LPS puede estar contenido en una unidad NAL cuya cabecera puede definirse de acuerdo con la Tabla 8 anterior. Los siguientes elementos sintácticos tienen la siguiente semántica ligeramente modificada cuando están asociadas con un LPS.

- 15 En este ejemplo, priority_id es igual al valor mínimo de los valores priority_id de todas las unidades NAL que se refieren a este LPS.

En este ejemplo, temporal_id es igual al valor mínimo del temporal_id de todas las unidades NAL que se refieren a este LPS.

- 20 En este ejemplo, dependency_id es igual al dependency_id de todas las unidades NAL que se refieren a este LPS.

En este ejemplo, quality_id es igual al valor mínimo de quality_id de todas las unidades NAL que se refieren a este LPS.

- 25 En este ejemplo, v_idx es el índice de vista del LPS actual. Todas las imágenes que se refieren a este LPS pueden tener un id de vista de view_id [v_idx].

- 30 De forma alternativa, los elementos sintácticos anteriores pueden señalarse directamente en la tabla de sintaxis del conjunto de parámetros de capa, como se muestra en el ejemplo de la Tabla 10. Se puede diseñar una tabla de sintaxis más detallada de acuerdo con la Tabla 9 siguiente. En este caso, esos elementos sintácticos no están en la cabecera de unidad NAL del LPS y el análisis del LPS puede depender del VPS con un ID igual a vps_id.

TABLA 10

35

layer_para_set() {	Descriptor
vps_id	u(8)
si(cnt_p > 1)	
priority_id	u(v)
si(cnt_t > 1)	
temporal_id	u(v)
reserved_one_bit	u(1)
si(cnt_d > 1)	
dependency_id	u(v)
si(cnt_q > 1)	
quality_id	u(v)
si(cnt_v > 1)	
view_idx	u(v)
depth_flag	u(1)
layer_para_set_id	ue(v)
// jerarquía de cu { {	
...	
rbsp_trailing_bits()	
}	

Un LPS, en este caso, no necesita tener una cabecera de unidad NAL duplicando los elementos sintácticos anteriores. Suponiendo que el tipo de unidad NAL de una unidad NAL que encapsula un LPS es, por ejemplo, 5, la

sintaxis de cabecera de unidad NAL puede modificarse ligeramente como se muestra en la Tabla 11, que añade la excepción "&& nal_unit_type != 5" en la instrucción "if" de la Tabla 8:

TABLA 11

5

nal_unit(NumBytesInNALunit) {	Descriptor
forbidden_zero_bit	f(1)
nal_ref_flag	u(1)
nal_unit_type	u(6)
NumBytesInRBSP = 0	
nalUnitHeaderBytes = 1	
si(nal_unit_type != 10 && nal_unit_type != 5) { // no unidad VPS NAL	
si(cnt_p > 1)	
priority_id	u(v)
si(cnt_t > 1)	
temporal_id	u(v)
reserved_one_bit	u(1)
si(cnt_d > 1)	
dependency_id	u(v)
si(cnt_q > 1)	
quality_id	u(v)
reserved_one_bit	u(1)
si(cnt_v > 1)	
view_idx	u(v)
si(cnt_f)	
reserved_flags	u(v)
m = Ceil(log2(cnt_p)) + Ceil(log2(cnt_t)) + Ceil(log2(cnt_d)) + Ceil(log2(cnt_q)) + Ceil(log2(cnt_v)) + cnt_f + 2	
si(((m + 7) >> 3) << 3) - m)	
reserved_bits	u(v)
nalUnitHeaderBytes += ((m + 7) >> 3)	
}	
para(i = nalUnitHeaderBytes; i < NumBytesInNALunit; i++) {	
si(i + 2 < NumBytesInNALunit && next_bits(24) == 0x000003) {	
rbsp_byte [NumBytesInRBSP++]	b(8)
rbsp_byte [NumBytesInRBSP++]	b(8)
i += 2	
emulation_prevention_three_byte /* igual a 0x03 */	f(8)
} si no	
rbsp_byte [NumBytesInRBSP++]	b(8)
}	
}	

En otros ejemplos, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden codificar los elementos sintácticos relacionados con las características escalables usando codificación de longitud fija, como se muestra en el ejemplo de la Tabla 12 siguiente.

10

TABLA 12

layer_para_set() {	Descriptor
---------------------	------------

vps_id	u(8)
priority_id	u(5)
temporal_id	u(3)
dependency_id	u(3)
quality_id	u(3)
view_idx	u(8)
layer_para_set_id	ue(v)
// jerarquía de cu { {	

La Tabla 13 siguiente proporciona un ejemplo de sintaxis para un conjunto de parámetros de imagen (PPS) de acuerdo con las técnicas de esta divulgación. En este ejemplo, el conjunto de parámetros de imagen no necesita señalar un "seq_parameter_set_id", contrariamente al PPS de HEVC convencional.

5

TABLA 13

pic_parameter_set_rbsp() {	Descriptor
pic_parameter_set_id	ue(v)
...	
pps_extension_flag	u(1)
si(pps_extension_flag)	
mientras(more_rbsp_data())	
pps_extension_data_flag	u(1)
rbp_trailing_bits()	
}	

A continuación se describen ejemplos de la semántica para el PPS de la Tabla 13.

10

En este ejemplo, pps_extension_flag igual a 0 especifica que no hay elementos sintácticos pps_extension_data_flag presentes en la estructura de sintaxis Rbsp del conjunto de parámetros de imagen. En este ejemplo, pps_extension_flag es igual a 0 en flujos de bits que cumplen con la próxima norma HEVC. El valor 1 para pps_extension_flag está reservado para un uso futuro por ITU-T | ISO/IEC. Los decodificadores, como el

15

En este ejemplo, pps_extension_data_flag puede tener cualquier valor. No afecta necesariamente a la conformidad con los perfiles especificados en la próxima norma HEVC. La semántica para valores de pps_extension_data_flag puede asignarse en desarrollos adicionales de la norma HEVC o ampliaciones de la norma sin entrar en conflicto con las técnicas de esta divulgación.

20

En las técnicas de esta divulgación, no es necesario señalar en el PPS ningún id del conjunto de parámetros de secuencia o id del conjunto de parámetros de capa. Algunos otros elementos sintácticos en PPS se pueden mover a LPS. Es decir, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden configurarse para codificar uno o más LPS incluyendo datos similares a los elementos sintácticos mostrados en la Tabla 13.

25

La FIG. 6 es un diagrama conceptual que ilustra un ejemplo de un conjunto de parámetros de agrupación (GPS) y relaciones del GPS con otros conjuntos de parámetros y cabeceras de fragmentos. En este ejemplo, los otros conjuntos de parámetros incluyen LPS, SPS, PPS, conjuntos de parámetros de adaptación (APS) de tipo 0 (por ejemplo, APS que señalan parámetros de filtro adaptativo de bucle (ALF)), APS de tipo 1 (por ejemplo, APS que señalan una matriz de cuantización) y otros conjuntos de parámetros. En este ejemplo, el GPS incluye una pluralidad de grupos diferentes, cada uno de los cuales tiene un ID de GPS único (también denominado ID de grupo), en el que cada grupo indica uno particular de cada uno de los diversos conjuntos de parámetros mediante el ID de un conjunto de parámetros. De esta manera, las cabeceras de fragmento solo necesitan especificar un group_id para especificar cada uno de los conjuntos de parámetros correspondientes al grupo que tiene ese group_id.

30

35

Las tablas 14 y 15 siguientes proporcionan ejemplos alternativos de sintaxis para un conjunto de parámetros de agrupación RBSP.

TABLA 14

5

group_para_set_rbsp() {	Descriptor
number signaled_para_set_groups_minus1	ue(v)
para(i = 0; i<= number_signaled_para_set_groups_minus1; i++) {	
para_set_group_id[i]	ue(v)
lps_id[i]	ue(v)
pps_id[i]	ue(v)
para (j= 0; j< numParaSetTypes; j++)	
para_set_type_id[i] [j]	
}	
gps_extension_flag	u(1)
si(gps_extension_flag)	
mientras(more_rbsp_data())	
gps_extension_data_flag	u(1)
rbp_trailing_bits()	
}	

TABLA 15

group_para_set_rbsp() {	Descriptor
number_signaled_para set_groups_minus1	ue(v)
para(i = 0; i<= number_signaled_para_set_groups_minus1; i++) {	
para_set_roup_id[i]	ue(v)
lps_id[i]	ue(v)
pps_id[i]	ue(v)
aps_id[i]	ue(v)
}	
gps_extension_flag	u(1)
si(gps_extension_flag)	
mientras(more_rbsp_data())	
gps_extension_data_flag	u(1)
rbp_trailing_bits()	
}	

10 Los codificadores de vídeo, tales como el codificador de vídeo 20 y el decodificador de vídeo 30, pueden configurarse para codificar un conjunto de parámetros de agrupación de acuerdo con, por ejemplo, la Tabla 14 o la Tabla 15. A continuación se proporcionan ejemplos de semántica para la sintaxis de los conjuntos de parámetros de agrupación.

15 En este ejemplo, number_signalled_para_set_groups_minus1 más 1 especifica el número de grupos de parámetros señalados. Este valor puede estar en la gama entre 0 y 30, inclusive.

En este ejemplo, para_set_group_id[i] especifica el ID del i-ésimo grupo de conjuntos de parámetros señalados. El valor de para_set_group_id[i] estará en la gama entre 0 y 31, inclusive.

20

En este ejemplo, para_set_type_id[i] [j] especifica el ID del j-ésimo tipo del conjunto de parámetros para el i-ésimo

grupo del conjunto de parámetros.

En este ejemplo, `lps_id [ i ]` indica el id del conjunto de parámetros de capa al que hace referencia el grupo de conjuntos de parámetros con un id de grupo de `para_set_group_id [ i ]`. Los valores de `dependency_id` y `view_idx` de un LPS con `layer_para_set_id` igual a `lps_id [ i ]` pueden ser idénticos a los valores de `dependency_id` y `view_idx`, respectivamente, de la unidad NAL del grupo de conjuntos de parámetros.

Los valores de `dependency_id` y `view_idx` de un conjunto de parámetros RBSP están presentes en la cabecera de unidad NAL de esta RBSP en los ejemplos de las Tablas 14 y 15 y los valores de `dependency_id` y `view_idx` de un LPS pueden estar presentes en la cabecera de unidad NAL de este LPS o en la tabla de sintaxis del LPS.

De forma alternativa, los valores de `dependency_id` y `view_idx` de un LPS con `layer_para_set_id` igual a `lps_id[i]` pueden no ser idénticos a los valores de `dependency_id` y `view_idx`, respectivamente, de la unidad NAL del grupo de conjuntos de parámetros.

En este ejemplo, `pps_id[ i ]` indica el id del conjunto de parámetros de imagen al que hace referencia el grupo de conjuntos de parámetros con un id de grupo de `para_set_group_id [ i ]`.

En este ejemplo, `aps_id [ i ]` indica el id del conjunto de parámetros de adaptación al que hace referencia el grupo de conjuntos de parámetros con un id de grupo de `para_set_group_id [ i ]`.

En este ejemplo, `gps_extension_flag` igual a 0 especifica que no hay elementos sintácticos `gps_extension_data_flag` presentes en la estructura de sintaxis RBSP de la agrupación de conjunto de parámetros. `Gps_extension_flag` puede ser igual a 0 en flujos de bits que cumplen con la próxima norma HEVC. El valor 1 para `gps_extension_flag` puede estar reservado para un uso futuro por parte de ITU-T | ISO/IEC. Los decodificadores, como el decodificador de vídeo 30, pueden ignorar todos los datos que siguen al valor 1 para `gps_extension_flag` en la unidad NAL de la agrupación del conjunto de parámetros. En general, `gps_extension_data_flag` puede tener cualquier valor. No tiene por qué afectar a la conformidad con los perfiles especificados en la próxima norma HEVC.

En algunos ejemplos, `para_set_type_id [ i ] [ j ]` puede en lugar de eso ser `aps_id [ i ] [ j ]`, con similar semántica `aaps_id [ i ]` como se ha descrito anteriormente.

Como se muestra en la FIG. 6, en lugar de referirse al ID del conjunto de parámetros de imagen en la cabecera de fragmento, de acuerdo con las técnicas de esta divulgación, la cabecera de fragmento puede referirse a un ID del grupo de conjuntos de parámetros, haciendo referencia indirectamente a un LPS, un PPS y un APS de cada tipo (por ejemplo, APS que proporcionan parámetros ALF y matrices de cuantización).

Un codificador de vídeo puede activar un conjunto de parámetros de vídeo o unos conjuntos de parámetros de capa cuando una unidad VCL NAL (que contiene un fragmento codificado) se refiere al conjunto de parámetros, indirectamente, por ejemplo, basándose en el principio de diseño H.264/AVC.

En algunos ejemplos, los conjuntos de parámetros pueden ser activados por un tipo específico de unidad NAL en lugar de por un fragmento codificado. Por ejemplo, un tipo de unidad NAL de este tipo específico (unidad de NAL de activación de conjuntos de parámetros), si está presente en el flujo de bits, puede activar uno y exactamente uno, VPS. En varias alternativas, además, tal tipo de unidad NAL puede activar al menos un LPS. Además, dicho tipo de unidad NAL puede activar al menos un PPS. Además, dicho tipo de unidad NAL puede activar al menos un APS. Una unidad NAL de activación de conjuntos de parámetros puede ser un conjunto de parámetros de agrupación RBSP. Una unidad NAL de activación de conjuntos de parámetros (PSA) puede ser aplicable a una secuencia de vídeo codificada. Una unidad PSA NAL puede considerarse una unidad NAL no VCL, es decir, no es directamente relevante para un codificador de vídeo. La sintaxis de cabecera de unidad NAL de la unidad PSA NAL puede ser la misma que una unidad VAL NAL.

En algunos ejemplos, una unidad PSA NAL, si está presente en una unidad de acceso, puede preceder a la primera unidad VAL NAL de la unidad de acceso. Puede haber al menos una unidad PSA NAL en la primera unidad de acceso de una secuencia de vídeo codificada, por ejemplo, una imagen IDR. Múltiples unidades PSA NAL en la misma secuencia de vídeo codificada pueden contener el mismo id de VPS; por lo tanto, no es necesario activar diferentes conjuntos de parámetros de vídeo dentro de la misma secuencia de vídeo codificada. Una unidad PSA NAL, si está presente en una unidad de acceso, puede preceder a cualquier unidad LPS, PPS, APS o SEI NAL, si está presente. Una unidad VPS NAL, si está presente en una unidad de acceso, puede preceder a cualquier unidad LPS, PPS, APS o SEI NAL, si está presente. En varias alternativas, además, una unidad PSA NAL, si está presente en una unidad de acceso, puede preceder a una unidad VPS NAL, si está presente.

En algunos ejemplos, los codificadores de vídeo, tales como el codificador de vídeo 20 y el decodificador de vídeo 30, pueden configurarse para utilizar la sintaxis de la Tabla 16 para un conjunto de parámetros de secuencia (SPS), en oposición a la sintaxis SPS convencional de, por ejemplo, la HEVC.

TABLA 16

rbasp_de_conjunto_de_parámetros_de_secuencia () {	Descriptor
video_parameter_set_id	ue(v)
rep_format_idx	ue(v)
pcm_enabled_flag	u(1)
si(pcm_enabled_flag) {	
pcm_bit_depth_luma_minus1	u(4)
pcm_bit_depth_chroma_minus1	u(4)
}	
qpprime_y_zero_transquant_bypass_flag	u(1)
log2_max_pic_order_cnt_lsb_minus4	ue(v)
para(i = 0; i <= max_temporal_layers_minus1; i++) {	
max_dec_pic_buffering[i]	ue(v)
num_reorder_pics[i]	ue(v)
max_latency_increaser[i]	ue(v)
}	
restricted_ref_pic_lists_flag	u(1)
si(restricted_ref_pic_lists_flag)	
lists_modification_present_flag	u(1)
log2_min_coding_block_size_minus3	ue(v)
log2_diff_max_min_coding_block_size	ue(v)
log2_min_transform_block_size_minus2	ue(v)
log2_diff_max_min_transform_block_size	ue(v)
si(pcm_enabled_flag) {	
log2_min_pcm_coding_block_size_minus3	ue(v)
log2_diff_max_min_pcm_coding_block_size	ue(v)
}	
max_transform_hierarchy_depth_inter	ue(v)
max_transform_hierarchy_depth_intra	ue(v)
scaling_list_enable_flag	
chroma_pred_from_luma_enabled_flag	u(1)
deblocking_filter_in_aps_enabled_flag	u(1)
seq_loop_filter_across_slices_enabled_flag	u(1)
asymmetric_motion_partitions_enabled_flag	u(1)
non_square_quadtree_enabled_flag	u(1)
sample_adaptive_offset_enabled_flag	u(1)
adaptive_loop_filter_enabled_flag	u(1)
si(adaptive_loop_filter_enabled_flag)	
alf_coef_in_slice_flag	u(1)
si(pcm_enabled_flag)	
pcm_loop_filter_disable_flag	u(1)
temporal_id_nesting_flag	u(1)
si(log2_min_coding_block_size_minus3 == 0)	
inter_4x4_enabled_flag	u(1)
long_term_ref_pics_present_flag	u(1)

tiles_or_entropy_coding_sync_idc	u(2)
si(tiles_or_entropy_coding_sync_idc == 1) {	
num_tile_columns_minus1	ue(v)
num_tile_rows_minus1	ue(v)
uniform_spacing_flag	u(1)
si(!uniform_spacing_flag) {	
para(i = 0; i < num_tile_columns_minus1; i++)	
column_width[i]	ue(v)
para(i = 0; i < num_tile_rows_minus1; i++)	
row_height[i]	ue(v)
}	
loop_filter_across_tiles_enabled_flag	u(1)
}	
vui_parameters_present_flag	u(1)
si(vui_parameters_present_flag)	
vui_parameters()	
sps_extension_flag	u(1)
si(sps_extension_flag)	
mientras(more_rbsp_data())	
sps_extension_data_flag	u(1)
rbsp_trailing_bits()	
}	

El ejemplo SPS de la Tabla 16 elimina profile_idc, reserved_zero_8bits, level_idc, chroma_format_idc, separate_colour_plane_flag y correspondiente condicional "si", max_temporal_layers_minus1, pic_width_in_luma_samples, pic_height_in_luma_samples, pic_cropping_flag, pic_crop_left_offset, pic_crop_right_offset, pic_crop_top_offset, y pic_crop_bottom_offset y la correspondiente instrucción condicional "si", bit_depth_luma_minus8, bit_depth_chroma_minus8, num_short_term_ref_pic_sets y short_term_ref_pic_set(i) y la correspondiente instrucción condicional "si" de la sintaxis SPS convencional. Además, el ejemplo SPS de la Tabla 16 añade un video_parameter_set_id y rep_format_idx. La semántica para los demás elementos sintácticos restantes puede ser la misma que se define en la HEVC convencional. La semántica para los elementos añadidos video_parameter_set_id y rep_format_idx se puede definir de la siguiente manera:

En este ejemplo, video_parameter_set_id identifica el conjunto de parámetros de vídeo (VPS) al que hace referencia el SPS actual. De forma alternativa, video_parameter_set_id no necesita ser señalado y puede usarse un GPS para vincular un SPS a un VPS específico.

En este ejemplo, rep_format_idx especifica el índice del formato de representación señalado en el conjunto de parámetros de vídeo a los que se hace referencia.

Como otra alternativa, la Tabla 17 proporciona otro ejemplo de sintaxis para un conjunto de parámetros de agrupación. Se supone que, en este ejemplo, el elemento sintáctico del ID de conjunto de parámetros de vídeo no está presente en la sintaxis de SPS, como se ha descrito anteriormente.

TABLA 17

group_para_set_rbsp() {	Descriptor
gps_is	ue(v)
vps_id	ue(v)
sps_id	ue(v)
pps_id	ue(v)
num_ref_aps_ids	ue(v)

para(i = 0; i < num_ref_aps_ids; i++) {	
ref_aps_id[i]	ue(v)
ref_aps_param_type[i]	ue(v)
}	
gps_extension_flag	u(1)
si(gps_extension_flag)	
mientras(more_rbsp_data())	
gps_extension_data_flag	u(1)
rbsp_trailing_bits()	
}	

La semántica para los elementos sintácticos de la Tabla 17 puede definirse de la forma siguiente:

En este ejemplo, **gps_id** especifica el identificador de un conjunto de parámetros de grupo (GPS).

5 En este ejemplo, **vps_id** especifica el identificador del conjunto de parámetros de vídeo al que hace referencia el GPS.

10 En este ejemplo, **sps_id** especifica el identificador del conjunto de parámetros de secuencia al que hace referencia el GPS.

En este ejemplo, **pps_id** especifica el identificador del conjunto de parámetros de secuencia de imágenes al que hace referencia el GPS.

15 En este ejemplo, **num_ref_aps_ids** especifica el número de los siguientes elementos sintácticos **ref_aps_id [i]**. El valor de **num_ref_aps_ids** puede estar en la gama entre 0 y 4, inclusive.

20 En este ejemplo, **ref_aps_id [i]** identifica el i-ésimo conjunto de parámetros de adaptación al que hace referencia el conjunto de parámetros de grupo.

El mismo valor de la **ref_aps_id [i]** puede estar presente en el bucle más de una vez y, por lo tanto, se puede hacer referencia a más de un tipo de parámetros APS de la misma APS por parte del mismo GPS y puede aplicarse a los fragmentos codificados en relación con el GPS .

25 En este ejemplo, **ref_aps_param_type [i]** especifica el tipo de los parámetros APS incluidos en el i-ésimo conjunto de parámetros de adaptación al que hace referencia el conjunto de parámetros de grupo. El valor de **ref_aps_param_type [i]** puede estar en la gama entre 0 y 3, inclusive. Los valores de 0 a 3, inclusive, para **ref_aps_param_type [i]** corresponden a los tipos de parámetro APS de la lista de escalado, filtro de desbloqueo, desviación de adaptación de muestra (SAO) y ALF, respectivamente. Los valores de **ref_aps_param_type [i]** para dos valores cualesquiera diferentes de i no serán idénticos, en algunos ejemplos.

30 En este ejemplo, **gps_extension_flag** igual a 0 especifica que no hay elementos sintácticos **gps_extension_data_flag** presentes en la estructura de sintaxis RBSP de agrupación del conjunto de parámetros. **gps_extension_flag** puede ser igual a 0 en flujos de bits que cumplen con la próxima norma HEVC. El valor 1 para **gps_extension_flag** puede estar reservado para un uso futuro por parte de ITU-T | ISO/IEC. Los descodificadores, como el descodificador de vídeo 30, pueden ignorar todos los datos que siguen al valor 1 para **gps_extension_flag** en la unidad NAL de la agrupación del conjunto de parámetros.

35 En este ejemplo, **gps_extension_data_flag** puede tener cualquier valor. No tiene por qué afectar la conformidad con los perfiles especificados en la próxima norma HEVC.

40 Los codificadores de vídeo, tales como el codificador de vídeo 20 y el descodificador de vídeo 30, pueden aplicar el siguiente proceso para activar conjuntos de parámetros para flujos de bits de una sola capa o de una sola vista, cuando el GPS se especifica de acuerdo con la Tabla 17 o se ajusta sustancialmente al ejemplo de la Tabla 17.

45 Una RBSP de un conjunto de parámetros de adaptación puede incluir parámetros a los que pueden referirse indirectamente las unidades NAL de fragmento codificado de una o más imágenes codificadas a través de uno o más conjuntos de parámetros de grupo a los que hacen referencia las unidades NAL de fragmento codificado. Cada RBSP de un conjunto de parámetros de adaptación puede considerarse inicialmente no activa al comienzo del

funcionamiento del proceso de decodificación. Como máximo una RBSP de un conjunto de parámetros de adaptación puede considerarse activa para cada tipo de parámetros APS en cualquier momento dado durante el funcionamiento del proceso de decodificación, y la activación de cualquier RBSP de un conjunto de parámetros de adaptación particular para un tipo particular de parámetros APS da como resultado la desactivación de la RBSP de un conjunto de parámetros de adaptación previamente activa (si existe) para ese tipo particular de parámetros APS.

Cuando una RBSP de un conjunto de parámetros de adaptación (con un valor particular de `aps_id`) no está activa para un tipo particular de parámetros de APS y se hace referencia a la misma indirectamente por parte de una unidad NAL de fragmento codificado para ese tipo particular de parámetros APS (usando ese valor de `aps_id`) a través de un conjunto de parámetros de grupo al que hace referencia la unidad NAL de fragmento codificado, puede activarse para ese tipo particular de parámetros APS. Esta RBSP del conjunto de parámetros de adaptación se denomina RBSP del conjunto de parámetros de adaptación activa para ese tipo particular de parámetros APS hasta que se desactiva por la activación de otra RBSP de un conjunto de parámetros de adaptación para ese tipo particular de parámetros APS. Una RBSP de un conjunto de parámetros de adaptación, con ese valor particular de `aps_id`, puede estar disponible para el proceso de decodificación antes de su activación.

Una RBSP de un conjunto de parámetros de imagen puede incluir parámetros a los que las unidades NAL de fragmento codificadas de una o más imágenes codificadas pueden referirse indirectamente a través de uno o más conjuntos de parámetros de grupo a los que hacen referencia las unidades NAL de fragmento codificado. Cada RBSP de un conjunto de parámetros de imagen puede considerarse inicialmente como activa al comienzo del funcionamiento del proceso de decodificación. A lo sumo una RBSP de un conjunto de parámetros de imagen puede considerarse activa en cualquier momento dado durante el funcionamiento del proceso de decodificación, y la activación de cualquier RBSP de un conjunto de parámetros de imagen específica da como resultado la desactivación de la RBSP de un conjunto de parámetros de imagen previamente activa (si la hubiera).

Cuando una RBSP de un conjunto de parámetros de imagen (con un valor particular de `pic_parameter_set_id`) no está activa y se hace referencia a la misma indirectamente por parte de una unidad NAL de fragmento codificado (utilizando ese valor de `pic_parameter_set_id`) a través de un conjunto de parámetros de grupo al que se hace referencia parte de la unidad NAL de fragmento codificado, puede activarse. Esta RBSP de un conjunto de parámetros de imagen se denomina RBSP del conjunto de parámetros de imagen activa hasta que se desactiva mediante la activación de otra RBSP de un conjunto de parámetros de imagen. Una RBSP de un conjunto de parámetros de imagen, con ese valor particular de `pic_parameter_set_id`, puede estar disponible para el proceso de decodificación antes de su activación.

Cualquier conjunto de parámetros de imagen NAL que contenga el valor de `pic_parameter_set_id` para la RBSP de conjunto de parámetros de imagen activa para una imagen codificada puede tener el mismo contenido que el de la RBSP de conjunto de parámetros de imagen activa para la imagen codificada a menos que siga la última unidad VCL NAL de la imagen codificada y preceda a la primera unidad VCL NAL de otra imagen codificada.

Una RBSP de un conjunto de parámetros de secuencia puede incluir parámetros a los que pueden referirse indirectamente las unidades NAL de fragmento codificado de una o más imágenes codificadas a través de uno o más conjuntos de parámetros de grupo a los que hacen referencia las unidades NAL de fragmento codificado o puede hacerse referencia a la misma por parte de una o más unidades SEI NAL que contengan un mensaje SEI de período de almacenamiento intermedio. Cada RBSP de un conjunto de parámetros de secuencia puede considerarse inicialmente no activa al comienzo del funcionamiento del proceso de decodificación. A lo sumo una RBSP de un conjunto de parámetros de secuencia puede considerarse activa en cualquier momento dado durante el funcionamiento del proceso de decodificación, y la activación de cualquier RBSP de un conjunto de parámetros de secuencia específica da como resultado la desactivación de la RBSP de un conjunto de parámetros de secuencia previamente activa (si la hubiera).

Cuando una RBSP de un conjunto de parámetros de secuencia (con un valor particular de `seq_parameter_set_id`) no está ya activa y se hace referencia a la misma indirectamente por parte de una unidad NAL de fragmento codificado a través de un conjunto de parámetros de grupo al que hace referencia la unidad NAL de fragmento codificado (utilizando ese valor de `seq_parameter_set_id`) o se hace referencia a la misma por parte de una unidad SEI NAL que contiene un mensaje SEI de período de almacenamiento intermedio (que utiliza ese valor de `seq_parameter_set_id`), puede ser activada. Esta RBSP del conjunto de parámetros de secuencia se denomina RBSP del conjunto de parámetros de secuencia activa hasta que se desactiva mediante la activación de otra RBSP de un conjunto de parámetros de secuencia. Una RBSP de un conjunto de parámetros de secuencia, con ese valor particular de `seq_parameter_set_id` y contenida dentro de una unidad de acceso con `temporal_id` igual a 0, puede estar disponible para el proceso de decodificación antes de su activación. Una RBSP de un conjunto de parámetros de secuencia activada permanecerá activa durante la secuencia entera de vídeo codificado.

Una RBSP de un conjunto de parámetros de vídeo puede incluir parámetros a los que pueden referirse indirectamente las unidades NAL de fragmento codificado de una o más imágenes codificadas a través de uno o más conjuntos de parámetros de grupo a los que hacen referencia las unidades NAL de fragmento codificado, o puede hacerse referencia a los mismos por parte de una o más unidades SEI NAL que contienen un mensaje SEI de

período de almacenamiento intermedio. Cada RBSP de un conjunto de parámetros de vídeo puede considerarse inicialmente no activa al comienzo del funcionamiento del proceso de decodificación. A lo sumo una RBSP de un conjunto de parámetros de vídeo puede considerarse activa en cualquier momento dado durante el funcionamiento del proceso de decodificación, y la activación de cualquier RBSP de un conjunto de parámetros de vídeo específica da como resultado la desactivación de la RBSP de un conjunto de parámetros de vídeo previamente activa (si la hubiera).

Cuando una RBSP de un conjunto de parámetros de vídeo (con un valor particular de `video_parameter_set_id`) no está ya activa y se hace referencia a la misma indirectamente por parte de una unidad NAL de fragmento codificado a través de un conjunto de parámetros de grupo al que hace referencia la unidad NAL de fragmento codificado (utilizando ese valor de `video_parameter_set_id`) o se hace referencia a la misma por parte de una unidad SEI NAL que contiene un mensaje SEI de período de almacenamiento intermedio (que usa ese valor de `video_parameter_set_id`), puede activarse. Esta RBSP del conjunto de parámetros de vídeo se denomina RBSP del conjunto de parámetros de vídeo activa hasta que se desactiva mediante la activación de otra RBSP del conjunto de parámetros de vídeo. Una RBSP de un conjunto de parámetros de vídeo, con ese valor particular de `video_parameter_set_id` y contenido dentro de una unidad de acceso con `temporal_id` igual a 0, estará disponible para el proceso de decodificación antes de su activación. Una RBSP de un conjunto de parámetros de vídeo activada permanecerá activa durante la secuencia entera de vídeo codificado.

Cualquier unidad de NAL de conjunto de parámetros de secuencia que contenga el valor de `seq_parameter_set_id` para la RBSP del conjunto de parámetros de secuencia activa para una secuencia de vídeo codificado puede tener el mismo contenido que el de la RBSP del conjunto de parámetros de secuencia activos para la secuencia de vídeo codificada, a menos que siga a la última unidad de acceso de la secuencia de vídeo codificada y preceda a la primera unidad de VCL NAL y a la primera unidad de SEI NAL que contiene un mensaje de SEI de período de almacenamiento intermedio (cuando esté presente) de otra secuencia de vídeo codificada.

Cualquier unidad de NAL del conjunto de parámetros de vídeo que contenga el valor de `video_parameter_set_id` para la RBSP del conjunto de parámetros de vídeo activa para una secuencia de vídeo codificada puede tener el mismo contenido que el de la RBSP de conjunto de parámetros de vídeo activa para la secuencia de vídeo codificada, a menos que siga a la última unidad de acceso de la secuencia de vídeo codificada y pueda preceder a la primera unidad de VCL NAL y a la primera unidad de SEI NAL que contiene un mensaje de SEI de período de almacenamiento intermedio (cuando esté presente) de otra secuencia de vídeo codificada.

Todas las restricciones que se expresan en la relación entre los valores de los elementos sintácticos (y los valores de las variables derivadas de esos elementos sintácticos) en conjuntos de parámetros de vídeo, conjuntos de parámetros de secuencia, conjuntos de parámetros de imagen y conjuntos de parámetros de adaptación y otros elementos sintácticos son expresiones de restricciones que pueden aplicarse solamente a los conjuntos de parámetros de vídeo activos, al conjunto de parámetros de secuencia activos, al conjunto de parámetros de imagen activos y al conjunto de parámetros de adaptación activos para cada tipo particular de parámetros APS. Si existe cualquier RBSP de un conjunto de parámetros de vídeo que no esté activada en el flujo de bits, sus elementos sintácticos pueden tener valores que se ajusten a las restricciones especificadas si se activaron por referencia en un flujo de bits que se ajusta de otro modo. Si existe cualquier RBSP de un conjunto de parámetros de secuencia que no esté activada en el flujo de bits, sus elementos sintácticos pueden tener valores que se ajusten a las restricciones especificadas si se activaron por referencia en un flujo de bits que se ajusta de otro modo. Si existe cualquier RBSP de un conjunto de parámetros de imagen que no esté nunca activada en el flujo de bits, sus elementos sintácticos pueden tener valores que se ajusten a las restricciones especificadas si se activaron por referencia en un flujo de bits que se ajusta de otro modo. Si existe cualquier RBSP de conjunto de parámetros de adaptación que no esté nunca activada en el flujo de bits, sus elementos sintácticos pueden tener valores que se ajusten a las restricciones especificadas si se activaron por referencia en un flujo de bits que se ajusta de otro modo.

Durante el funcionamiento del proceso de decodificación, pueden considerarse en efecto los valores de los parámetros del conjunto de parámetros de vídeo activos, el conjunto de parámetros de secuencia activos, el conjunto de parámetros de imagen activos y el conjunto de parámetros de adaptación activos para cada tipo de parámetros APS. Para la interpretación de los mensajes SEI, pueden considerarse en efecto los valores de los parámetros del conjunto de parámetros de vídeo, conjunto de parámetros de secuencia, conjunto de parámetros de imagen y conjunto de parámetros de adaptación que están activos para el funcionamiento del proceso de decodificación para las unidades VCL NAL de la imagen codificada en la misma unidad de acceso a menos que se especifique lo contrario en la semántica de mensajes SEI.

La FIG. 7 es un diagrama de flujo que ilustra un ejemplo de procedimiento para la codificación de datos de vídeo de acuerdo con las técnicas de esta divulgación. Aunque se describe con respecto al codificador de vídeo 20, debe entenderse que otros dispositivos de codificación de vídeo pueden configurarse para llevar a cabo el procedimiento de la FIG. 7.

Inicialmente, en este ejemplo, el codificador de vídeo 20 recibe un flujo de bits que incluye una o más capas de datos de vídeo sin procesar (100). Por ejemplo, la fuente de vídeo 18 (FIG. 1) puede proporcionar datos de vídeo

multivista al codificador de vídeo 20. De forma alternativa, el codificador de vídeo 20, o un preprocesador del mismo, puede dividir un flujo de bits de vídeo sin procesar en una pluralidad de diversas capas, por ejemplo, capas de resolución espacial, capas de calidad, capas temporales o similares. En otros ejemplos más, un flujo de bits puede dividirse en una combinación de varias capas, por ejemplo, cualquier combinación de vistas, capas de resolución espacial, capas de calidad, capas temporales o similares.

El codificador de vídeo 20 puede determinar uno o más parámetros comunes para secuencias correspondientes entre un conjunto de capas (102). Las secuencias correspondientes pueden ser secuencias que tienen posiciones temporales correspondientes en diferentes capas. Es decir, una primera secuencia, que tiene un tiempo de inicio (en términos de tiempo de visualización) de T1 y un tiempo de finalización (de nuevo en términos de tiempo de visualización) de T2, y una segunda secuencia, que también tiene un tiempo de inicio de T1 y un tiempo de terminación de T2, se puede decir que se corresponden entre sí. En particular, la primera secuencia puede formar parte de una primera capa y la segunda secuencia puede formar parte de una segunda capa diferente. Una "secuencia" puede incluir una serie de imágenes consecutivas en orden de descodificación, por ejemplo, comenzando con una imagen de actualización de descodificación instantánea (IDR) y terminando inmediatamente antes de una imagen IDR posterior en el orden de descodificación. En general, los parámetros pueden corresponder a un conjunto de secuencias correspondientes de una o más capas, por ejemplo, N capas, donde N es un número entero. El codificador de vídeo 20 puede entonces codificar un VPS que incluye datos para los parámetros determinados (104). Por ejemplo, el codificador de vídeo 20 puede codificar un VPS correspondiente a uno de los ejemplos de la Tabla 2 o de la Tabla 5.

El codificador de vídeo 20 también puede determinar parámetros comunes para una secuencia dentro de una capa (106). La secuencia puede comprender una de las secuencias correspondientes a otras secuencias en otras capas para las que se codificó el VPS. El codificador de vídeo 20 puede codificar un conjunto de parámetros de secuencia (SPS) que incluye los parámetros comunes para la secuencia (108). Por lo tanto, debe entenderse que el VPS y el SPS son estructuras de datos separadas y que corresponden a diferentes tipos de datos de vídeo. Mientras que un VPS puede corresponder a un conjunto de secuencias correspondientes entre una pluralidad de capas, el SPS corresponde a una secuencia en una capa. El SPS puede ajustarse sustancialmente a un SPS de H.264/AVC, el SPS de H.264/AVC ampliado por MVC (ilustrado en la Tabla 1 anterior), la próxima norma HEVC, o el ejemplo de la Tabla 16 descrito anteriormente. Además, el codificador de vídeo 20 puede codificar un conjunto de parámetros de imagen (PPS) para una imagen en la secuencia (110). El PPS puede ajustarse sustancialmente a un SPS de H.264/AVC, la próxima norma HEVC, o el ejemplo de la Tabla 13 descrito anteriormente. Aunque el procedimiento de la FIG. 7 muestra la codificación de solo un PPS, debe entenderse que se pueden codificar múltiples PPS. Una o más imágenes pueden referirse al mismo PPS.

El codificador de vídeo 20 puede determinar entonces si la capa reciente para la que se codificó un SPS y un PPS es la última capa (112). Si la última capa todavía no se ha tratado (ramificación "NO" de 112), el codificador de vídeo 20 puede seleccionar una capa siguiente y codificar un SPS y uno o más PPS para la siguiente capa, por ejemplo, de acuerdo con los pasos 106-110. Después de que se haya tratado la última capa (ramificación "SI" de 112), el codificador de vídeo 20 puede codificar datos de vídeo de las diversas capas basándose en los datos de los VPS, SPS y PPS. Diversos ejemplos de codificación de datos de vídeo basados, al menos en parte, en un VPS se describen con mayor detalle a continuación con respecto a las FIGs. 9-12.

Aunque no se muestra en el ejemplo de la FIG. 7, en algunos ejemplos, el codificador de vídeo 20 puede codificar adicionalmente uno o más LPS y/o uno o más GPS, como se ha descrito anteriormente. Los LPS pueden ajustarse sustancialmente a los ejemplos de la Tabla 9, Tabla 10 o Tabla 12, mientras que el GPS puede ajustarse sustancialmente a los ejemplos de la Tabla 14, Tabla 15 o Tabla 17. En tales ejemplos, el codificador de vídeo 20 codifica los datos de vídeo también basándose, al menos en parte, en los LPS y/o los GPS.

De esta manera, el procedimiento de la FIG. 7 representa un ejemplo de un procedimiento que incluye codificar un conjunto de parámetros de vídeo (VPS) para una o más capas de datos de vídeo, en el que cada una de una o más capas de datos de vídeo se refiere al VPS, y codificar una o más capas de datos de vídeo basándose, al menos en parte, en el VPS.

La FIG. 8 es un diagrama de flujo que ilustra un ejemplo de procedimiento de descodificación de datos de vídeo de acuerdo con las técnicas de esta divulgación. Aunque se describe con respecto al descodificador de vídeo 30, debe entenderse que otros dispositivos de descodificación de vídeo pueden configurarse para llevar a cabo el procedimiento de la FIG. 8.

Inicialmente, el descodificador de vídeo 30 recibe un flujo de bits que incluye un VPS, uno o más SPS, y uno o más PPS para capas de datos de vídeo codificados (120). El descodificador de vídeo 30 puede entonces descodificar el VPS, que incluye parámetros comunes para secuencias correspondientes entre una o más capas (122). Del mismo modo, el descodificador de vídeo 30 puede descodificar un conjunto de parámetros de secuencia que incluye parámetros comunes para una secuencia de una capa (124). Además, el descodificador de vídeo 30 puede descodificar un conjunto de parámetros de imagen que incluye parámetros para una imagen de la secuencia (126). Como se ha analizado anteriormente, una o más imágenes pueden referirse al mismo PPS y, por lo tanto, los

parámetros del PPS pueden considerarse comunes a una o más imágenes. Del mismo modo, el descodificador de vídeo 30 puede descodificar una pluralidad de PPS para la secuencia, aunque no se muestra en la FIG. 8.

Además, el descodificador de vídeo 30 puede determinar si la capa más reciente era la última capa a tratar (128). Si la capa más reciente no era la última capa (ramificación "NO" de 128), el descodificador de vídeo 30 puede proceder a descodificar un SPS y uno o más PPS para una capa posterior de acuerdo con las etapas 124 y 126. Por otra parte, si la capa más reciente era la última capa (ramificación "YES" de 128), el descodificador de vídeo 30 puede proceder a descodificar datos de vídeo de las capas basándose en los VPS, SPS y PPS (130). Ejemplos de datos de vídeo de codificación basados, al menos en parte, en un VPS se analizan con mayor detalle con respecto a las FIGs. 9-12.

Aunque no se muestra en el ejemplo de la FIG. 8, en algunos ejemplos, el descodificador de vídeo 30 puede descodificar adicionalmente uno o más LPS y/o uno o más GPS, como se ha descrito anteriormente. Los LPS pueden ajustarse sustancialmente a los ejemplos de la Tabla 9, Tabla 10 o Tabla 12, mientras que el GPS puede ajustarse sustancialmente a los ejemplos de la Tabla 14, Tabla 15 o Tabla 17. En tales ejemplos, el descodificador de vídeo 30 descodifica los datos de vídeo basándose también, al menos en parte, en los LPS y/o los GPS.

De esta manera, el procedimiento de la FIG. 8 representa un ejemplo de un procedimiento que incluye codificar un conjunto de parámetros de vídeo (VPS) para una o más capas de datos de vídeo, en el que cada una de una o más capas de datos de vídeo se refiere al VPS y codifica una o más capas de datos de vídeo basándose al menos en parte en el VPS.

La FIG. 9 es un diagrama de flujo que ilustra un ejemplo de procedimiento de codificación de datos de vídeo basándose, al menos en parte, en un cierto número de capas temporales como se señala en un VPS. El procedimiento de la FIG. 9 puede ser realizado por el codificador de vídeo 20 y/o el descodificador de vídeo 30. Para los propósitos del ejemplo, el procedimiento de la FIG. 9 se describe con respecto al descodificador de vídeo 30.

En este ejemplo, el descodificador de vídeo 30 codifica (es decir, descodifica) un VPS que indica un cierto número de capas temporales en datos de vídeo (150), por ejemplo, de una o más capas a las que corresponde el VPS. Por ejemplo, el descodificador de vídeo 30 puede descodificar "cnt_t" como se describe con respecto a la Tabla 2 anterior. Para mencionar otro ejemplo, el descodificador de vídeo 30 puede descodificar num_temporal_layers_minus1, como se describe con respecto a la Tabla 5 anterior.

Basándose en esta indicación, en este ejemplo, el descodificador de vídeo 30 descodifica identificadores temporales para cada una de las capas temporales (152). Del mismo modo, el descodificador de vídeo 30 puede determinar los valores de identificador de la imagen de referencia basándose en el número de capas temporales (154). Por ejemplo, el descodificador de vídeo 30 puede configurarse para determinar que, para una imagen actual en la capa N, la imagen actual no usará imágenes en o sobre la capa N+1 como referencia. Por lo tanto, el descodificador vídeo 30 puede determinar identificadores para posibles imágenes de referencia en capas en o debajo de la capa N. Además, el descodificador de vídeo 30 puede descodificar datos de imágenes en la capa temporal N usando datos de referencia de capas hasta (e incluyendo) la capa N (156). Por lo tanto, la FIG. 9 representa un ejemplo de un procedimiento que incluye codificación de datos de un VPS indicativo de un número máximo de capas temporales en una o más capas de datos de vídeo y codificación de una o más capas basándose, al menos en parte, en el VPS.

La FIG. 10 es un diagrama de flujo que ilustra un ejemplo de procedimiento de codificación de datos de vídeo basándose, al menos en parte, en un cierto número de imágenes a reordenar en una o más capas e imágenes a almacenar en una memoria intermedia de imágenes descodificadas. El procedimiento de la FIG. 10 puede ser realizado por el codificador de vídeo 20 y/o descodificador de vídeo 30. Para los propósitos del ejemplo, el procedimiento de la FIG. 10 se describe con respecto al descodificador de vídeo 30.

En este ejemplo, el descodificador de vídeo 30 descodifica un VPS que indica un cierto número de imágenes a reordenar en una o más capas de datos de vídeo y un cierto número de imágenes a almacenar en una memoria intermedia de imágenes descodificadas (por ejemplo, memoria de imágenes de referencia 82) en un momento dado (160). Por ejemplo, el descodificador de vídeo 30 puede descodificar un elemento sintáctico del VPS que corresponde sustancialmente a num_reorder_pics como se describe con respecto a la Tabla 16 anterior y/o información de restricción del flujo de bits que especifica un tamaño de DPB. En otros ejemplos, el VPS podría incluir solamente uno u otro, y no necesariamente ambos, del número de imágenes a reordenar y del número de imágenes que se van a almacenar en la memoria intermedia de imágenes descodificadas. El descodificador de vídeo 30 puede entonces gestionar la memoria intermedia de imágenes descodificadas (por ejemplo, la memoria de imágenes de referencia 82) basándose en el número de imágenes que se van a reordenar y/o almacenar (162). Por ejemplo, el descodificador de vídeo 30 puede eliminar imágenes de la memoria 82 de la imagen de referencia cuando se almacena más que el número de imágenes a almacenar en la memoria 82 de imágenes de referencia.

El descodificador de vídeo 30 también puede determinar los valores de identificador de la imagen de referencia basándose en el número de imágenes en la DPB (es decir, en la memoria de imágenes de referencia 82) (164). Además, el descodificador de vídeo 30 puede descodificar datos de imágenes basándose en los valores de

identificador de la imagen de referencia (166). De este modo, el procedimiento de la FIG. 10 representa un ejemplo de un procedimiento que incluye la codificación de datos de un VPS indicativo de un cierto número de imágenes a almacenar en una memoria intermedia de imágenes descodificadas (DPB) durante la descodificación de una o más capas y un procedimiento que incluye la codificación de datos de un VPS indicativo de un cierto número de tramas a reordenar en al menos una de una o más capas.

La FIG. 11 es un diagrama de flujo que ilustra un ejemplo de procedimiento de codificación de datos de vídeo basándose, al menos en parte, en parámetros de descodificador de referencia hipotéticos (HRD) señalados en un VPS. El procedimiento de la FIG. 11 puede ser realizado por el codificador de vídeo 20 y/o el descodificador de vídeo 30. Para los propósitos del ejemplo, el procedimiento de la FIG. 11 se describe con respecto al descodificador de vídeo 30.

En este ejemplo, el descodificador de vídeo 30 descodifica un VPS que indica los parámetros HRD (170). El descodificador de vídeo 30 puede determinar adicionalmente los tiempos de extracción para las imágenes de una memoria intermedia de imágenes codificadas (CPB) basándose en los parámetros HRD (172). El descodificador de vídeo 30 puede a continuación eliminar datos de la CPB basándose en los tiempos de eliminación determinados (174), y descodificar los datos eliminados de la CPB. Del mismo modo, el procedimiento de la FIG. 11 representa un ejemplo de un procedimiento que incluye la codificación de datos de un VPS indicativo de uno o más parámetros de descodificador de referencia hipotético (HRD) y codificación de datos de una o más capas basándose en los parámetros de HDR.

La FIG. 12 es un diagrama de flujo que ilustra un ejemplo de procedimiento de codificación de datos de vídeo basándose, al menos en parte, en datos de ampliación señalados en un VPS. El procedimiento de la FIG. 12 puede ser realizado por el codificador de vídeo 20 y/o descodificador de vídeo 30. Para los propósitos del ejemplo, el procedimiento de la FIG. 12 se describe con respecto al descodificador de vídeo 30.

El descodificador de vídeo 30, en este ejemplo, descodifica datos de un VPS indicando si el VPS incluye datos de ampliación (180). Por ejemplo, el descodificador de vídeo 30 puede descodificar un `vps_extension_flag` del VPS. El descodificador de vídeo 30 determina entonces si los datos indican que el VPS incluye datos de ampliación (182). Si los datos indican que el VPS incluye datos de ampliación (ramificación "SI" de 182), el descodificador de vídeo 30 codifica datos de ampliación VPS para una o más herramientas de codificación de ampliación (184) y descodifica datos de vídeo usando las herramientas de codificación de ampliación y datos de ampliación 186). Por otra parte, si los datos indican que el VPS no incluye datos de ampliación (ramificación "NO" de 182), el descodificador de vídeo 30 puede descodificar los datos de vídeo usando herramientas de codificación convencionales (188). De esta manera, el procedimiento de la FIG. 12 representa un ejemplo de un procedimiento que incluye codificación de datos de un VPS indicativo de si el VPS incluye una ampliación más allá de una norma correspondiente y cuando el VPS incluye la ampliación, datos para la ampliación, así como codificación de datos de vídeo basándose en los datos de ampliación del VPS.

Debe reconocerse que, dependiendo del ejemplo, ciertos actos o sucesos de cualquiera de las técnicas descritas en el presente documento pueden realizarse en una secuencia distinta, pueden añadirse, fundirse u omitirse por completo (por ejemplo, no todos los actos o sucesos descritos son necesarios para la puesta en práctica de las técnicas). Además, en ciertos ejemplos, los actos o sucesos pueden realizarse simultáneamente, por ejemplo, mediante el procesamiento de múltiples hebras, el procesamiento de interrupciones o múltiples procesadores, en lugar de secuencialmente.

En uno o más ejemplos, las funciones descritas pueden implementarse en hardware, software, firmware o cualquier combinación de los mismos. Si se implementan en software, las funciones, como una o más instrucciones o código, pueden almacenarse en, o transmitirse por, un medio legible por ordenador, y ejecutarse mediante una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que corresponden a un medio tangible tal como medios de almacenamiento de datos o medios de comunicación que incluyen cualquier medio que facilite la transferencia de un programa informático desde un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por ordenador pueden corresponder, en general, a (1) medios de almacenamiento tangibles y legibles por ordenador, que sean no transitorios, o (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser cualesquiera medios disponibles a los que se puede acceder desde uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para implementar las técnicas descritas en la presente divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

A modo de ejemplo, y no de limitación, tales medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento de disco óptico, almacenamiento de disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que pueda utilizarse para almacenar código de programa deseado en forma de instrucciones o estructuras de datos y al que pueda accederse mediante un ordenador. Además, cualquier conexión recibe adecuadamente la denominación de medios legibles por ordenador. Por ejemplo, si las instrucciones se transmiten desde una sede de la Red, un servidor u otra fuente

remota usando un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado digital (DSL) o tecnologías inalámbricas tales como infrarrojos, radio y microondas, entonces el cable coaxial, el cable de fibra óptica, el par trenzado, la DSL o las tecnologías inalámbricas tales como infrarrojos, radio y microondas se incluyen en la definición de medio. Sin embargo, debería entenderse que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino que, en cambio, se orientan a medios de almacenamiento tangibles no transitorios. Los discos, como se usan en el presente documento, incluyen el disco compacto (CD), el disco de láser, el disco óptico, el disco versátil digital (DVD), el disco flexible y el disco Blu-ray, donde algunos discos normalmente reproducen datos de manera magnética, mientras que otros discos reproducen los datos de manera óptica con láser. Las combinaciones de lo anterior también deberían incluirse dentro del alcance de los medios legibles por ordenador.

Las instrucciones pueden ser ejecutadas por uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados específicos de la aplicación (ASIC), formaciones lógicas programables en el terreno (FPGA) u otros circuitos lógicos integrados o discretos equivalentes. Por consiguiente, el término "procesador", como se usa en el presente documento, puede referirse a cualquier estructura anterior o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento puede proporcionarse dentro de módulos de hardware y/o software específicos configurados para la codificación y la decodificación, o incorporarse en un códec combinado. Además, las técnicas podrían implementarse completamente en uno o más circuitos o elementos lógicos.

En otros ejemplos más, esta divulgación contempla un medio legible por ordenador que comprende una estructura de datos almacenada en el mismo, en el que la estructura de datos incluye un flujo de bits codificado consistente con esta divulgación. En particular, el flujo de bits codificado puede incluir una o más capas de datos de vídeo y un parámetro de vídeo (VPS) para las una o más capas de datos de vídeo, en el que cada una de las una o más capas de datos de vídeo se refiere al VPS y una o más capas de datos de vídeo se codifican basándose, al menos en parte, en el VPS.

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

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

REIVINDICACIONES

1. Un procedimiento de codificación de datos de vídeo, comprendiendo el procedimiento:
 - 5 codificación (104, 122) de un conjunto de parámetros de vídeo, VPS, para una pluralidad de capas de datos de vídeo, en el que cada una de la pluralidad de capas de datos de vídeo se refiere al VPS, y en la que la codificación del VPS comprende:
 - 10 codificación de datos del VPS indicativos de un cierto número de tramas a reordenar en al menos una entre la pluralidad de capas de datos de vídeo,
 - codificación de datos del VPS indicativos de un cierto número de imágenes a almacenar en una memoria intermedia de imágenes descodificadas, DPB, durante la descodificación de la pluralidad de capas de datos de vídeo, y
 - 15 codificación de datos del VPS indicativos de un número máximo de capas temporales en la pluralidad de capas de datos de vídeo; y
 - codificación de (114, 130) la pluralidad de capas de datos de vídeo basándose, al menos en parte, en el VPS.
- 20 2. El procedimiento según la reivindicación 1, en el que la codificación del VPS comprende además al menos uno entre:
 - codificación de datos de los VPS indicativos de uno o más conjuntos de descodificadores de referencia hipotéticos, HRD, parámetros; o
 - 25 codificación de datos del VPS indicativos de si el VPS incluye una ampliación más allá de una norma correspondiente, y cuando el VPS incluye la ampliación, datos para la ampliación.
- 30 3. El procedimiento según la reivindicación 1, en el que la codificación de la pluralidad de capas de datos de vídeo comprende al menos uno entre:
 - codificación de la pluralidad de capas de datos de vídeo de acuerdo con la Codificación de Vídeo de Alta Eficiencia, HEVC; o
 - 35 codificación de la pluralidad de capas de datos de vídeo de acuerdo con, al menos uno entre, Codificación de Vídeo multivista, MVC y Codificación de Vídeo Escalable, SVC.
- 40 4. El procedimiento según la reivindicación 1, en el que codificar el VPS comprende codificar información que especifica, para una o más dimensiones de la pluralidad de capas de datos de vídeo, una o más de un cierto número de capas de prioridad en la pluralidad de capas de datos de vídeo, un cierto número de capas de dependencia en la pluralidad de capas de datos de vídeo, un cierto número de capas temporales en la pluralidad de capas de datos de vídeo, un número máximo de capas de calidad para cualquiera de las capas de dependencia en la pluralidad de capas de datos de vídeo y un número máximo de vistas en la pluralidad de capas de datos de vídeo y, preferentemente, en el que cuando un subconjunto de la pluralidad de capas de datos de vídeo tiene la misma resolución espacial y la misma profundidad de bits, cada una de las capas del subconjunto corresponde a una diferente de las capas de dependencia.
- 45 5. El procedimiento según la reivindicación 4, en el que codificar el VPS comprende codificar información que define una asignación de un índice de muestra a un indicador de características, y en el que la codificación de la información que define la asignación comprende al menos uno entre:
 - 50 codificación de información que especifica un indicador de características respectivo para cada uno de una pluralidad de índices de características cuando un indicador de características que define características de una dimensión de la pluralidad de capas de datos de vídeo no está dentro de un rango de índice de cero a un contador de dimensión de muestra menos 1 en el que el contador se define mediante un índice; o
 - 55 codificación de una o más de una resolución espacial respectiva para cada uno de una pluralidad de índices de dependencia, una velocidad de tramas para cada uno de una pluralidad de índices temporales, un identificador de vista para cada uno de una pluralidad de índices de vista, un par de valores de profundidad específicos para la luminancia y crominancia para cada uno de una pluralidad de índices de profundidad de bits, y un indicador de formato de muestreo de crominancia específico para cada uno de una pluralidad de formatos de muestreo de crominancia.
 - 60
- 65 6. El procedimiento según la reivindicación 1, en el que codificar el VPS comprende codificar información que define parámetros de control y uno o más indicadores de habilitación / deshabilitación de herramienta y

- preferentemente, en el que los parámetros de control y el uno o más indicadores de habilitación / deshabilitación de herramienta comprenden uno o más de `pcm_bit_depth_luma_minus1`, un `pcm_bit_depth_chroma_minus1`, un `loop_filter_across_slice_flag`, un `pcm_loop_filter_disable_flag`, un `temporal_id_nesting_flag`, uno o más elementos sintácticos relacionados con componentes, un `chroma_pred_from_luma_enabled_flag`, un `sample_adaptive_offset_enabled_flag`, un `adaptive_loop_filter_enabled_flag`, y un `inter_4x4_enabled_flag`.
- 5
7. El procedimiento según la reivindicación 1, en el que codificar el VPS comprende codificar información que define uno o más descriptores de punto de operación y, preferentemente, en el que codificando la información que define el uno o más descriptores de punto de operación comprende información de codificación que define uno o más de un cierto número de puntos de operación máxima, dependencia entre diferentes capas o vistas, perfil y nivel para cada uno de los puntos de operación, para cada punto de operación, capa de codificación de vídeo de punto de operación, VCL, capa de abstracción de red, NAL, representación de unidad, para cada dimensión uno o más de un valor de índice específico, una gama de posibles valores de índice para la dimensión, y una lista de valores de índice, velocidad de bits para cada uno de los puntos de operación, dependencia entre los puntos de operación, restricciones para cada uno de los puntos de operación, información de usabilidad de vídeo, VUI, para cada uno de los puntos de operación, y VUI para cada una de la pluralidad de capas de datos de vídeo.
- 10
- 15
8. El procedimiento según la reivindicación 1, que comprende además codificar un respectivo conjunto de parámetros de secuencia de capas, LPS, para cada una de la pluralidad de capas de datos de vídeo, en el que codificar la pluralidad de capas de datos de vídeo basándose al menos en parte en el VPS comprende codificar la pluralidad de capas de datos de vídeo basándose al menos en parte en el VPS y el LPS respectivo y, preferentemente, en el que la codificación del LPS respectivo para cada una de la pluralidad de capas de datos de vídeo comprende al menos uno de entre:
- 20
- 25
- codificación de información que define una indicación de dimensión de muestra que indica, para cada dimensión, un índice para cada dimensión;
- 30
- codificación de información que define los parámetros de control y los indicadores de habilitación / deshabilitación de herramientas y, preferentemente, en el que los parámetros de control y el uno o más indicadores de habilitación / deshabilitación de herramientas comprenden uno o más de un `pcm_bit_depth_luma_minus1`, un `pcm_bit_depth_chroma_minus1`, un `loop_filter_across_slice_flag`, un `pcm_loop_filter_disable_flag`, uno o más elementos sintácticos relacionados con componentes, un `chroma_pred_from_luma_enabled_flag`, un `sample_adaptive_offset_enabled_flag`, un `adaptive_loop_filter_enabled_flag`, y una unidad de codificación, CU, jerarquía; o
- 35
- codificación de información que define la información de uno o más conjuntos de parámetros diferentes que se aplican a al menos uno de un fragmento, un grupo de fragmentos, una imagen y varias imágenes que se refieren a un conjunto de parámetros de imagen común, PPS.
- 40
9. El procedimiento según la reivindicación 1, que comprende además codificar uno o más conjuntos de parámetros de imagen, PPS de tal manera que los PPS no se refieren al VPS, y no se refieren a conjuntos de parámetros de secuencia de capas, LPS de la pluralidad de capas de datos de vídeo y, preferiblemente, en el que la codificación de la pluralidad de capas de datos de vídeo basándose al menos en parte en el VPS comprende codificar la pluralidad de capas de datos de vídeo basándose al menos en parte en el VPS, los PPS y los LPS, de tal manera que cuando un elemento sintáctico de uno de los PPS entra en conflicto con el VPS o uno respectivo de los LPS, codificar una correspondiente de la pluralidad de capas de datos de vídeo basándose en el elemento sintáctico de uno de los PPS.
- 45
- 50
10. El procedimiento según la reivindicación 1, que comprende además la codificación de un conjunto de parámetros de agrupación, GPS, que agrupa todos los conjuntos de parámetros, incluyendo el VPS, para la pluralidad de capas de datos de vídeo juntas y preferentemente, en el que codificar el GPS comprende la codificación de información que define un identificador del GPS, comprendiendo además el procedimiento la codificación de información de una cabecera de fragmento correspondiente al identificador del GPS.
- 55
11. El procedimiento según la reivindicación 1, en el que la codificación de la pluralidad de capas de datos de vídeo comprende descodificar la pluralidad de capas de datos de vídeo, y en el que la codificación del VPS comprende analizar el VPS.
- 60
12. El procedimiento según la reivindicación 1, en el que la codificación de la pluralidad de capas de datos de vídeo comprende codificar la pluralidad de capas de datos de vídeo, y en el que codificar el VPS comprende construir el VPS.
- 65
13. Un dispositivo para codificar datos de vídeo, comprendiendo el dispositivo:

medios para codificar (20, 30) un conjunto de parámetros de vídeo, VPS, para una pluralidad de capas de datos de vídeo, en el que cada una de la pluralidad de capas de datos de vídeo se refiere al VPS y en el que los medios para codificar el VPS comprenden:

- 5 medios para codificar datos del VPS indicativos de un cierto número de tramas a reordenar en al menos una entre la pluralidad de capas de datos de vídeo,
- medios para codificar datos del VPS indicativos de un cierto número de imágenes a almacenar en una memoria intermedia de imágenes descodificadas, DPB, durante la descodificación de la pluralidad de capas de datos de vídeo, y
- 10 medios para codificar datos del VPS indicativos de un número máximo de capas temporales en la pluralidad de capas de datos de vídeo; y

medios para codificar (20, 30) la pluralidad de capas de datos de vídeo basándose al menos en parte en el VPS.

- 15 **14.** El dispositivo según la reivindicación 13, en el que los medios para codificar el VPS comprenden además al menos uno entre:

- 20 medios para codificar datos del VPS indicativos de uno o más conjuntos de parámetros de descodificador hipotético de referencia, HRD; o

- 25 medios para codificar datos del VPS indicativos de si el VPS incluye una ampliación más allá de una norma correspondiente, y cuando el VPS incluya la ampliación, datos para la ampliación y en el que los medios para codificar la pluralidad de capas de datos de vídeo comprenden medios para codificar la pluralidad de capas de datos de vídeo de acuerdo con uno de Codificación de Vídeo de Alta Eficiencia, HEVC, Codificación de Vídeo multivista, MVC y Codificación de Vídeo Escalable, SVC.

- 15.** Un medio de almacenamiento legible por ordenador que tiene instrucciones almacenadas en el mismo que, cuando son ejecutadas, hacen que un procesador realice el procedimiento según cualquiera de las reivindicaciones 1 -12.
- 30

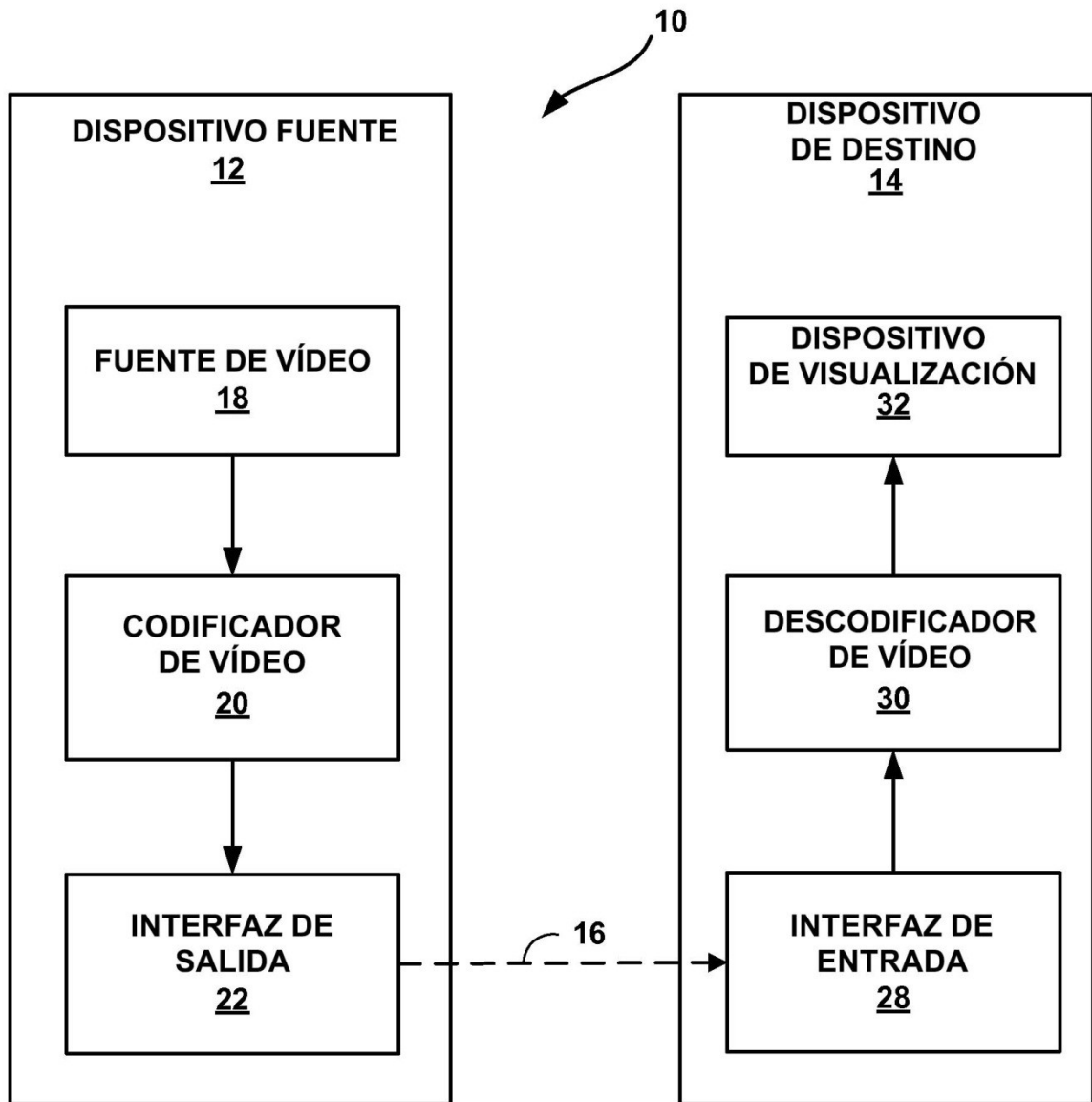


FIG. 1

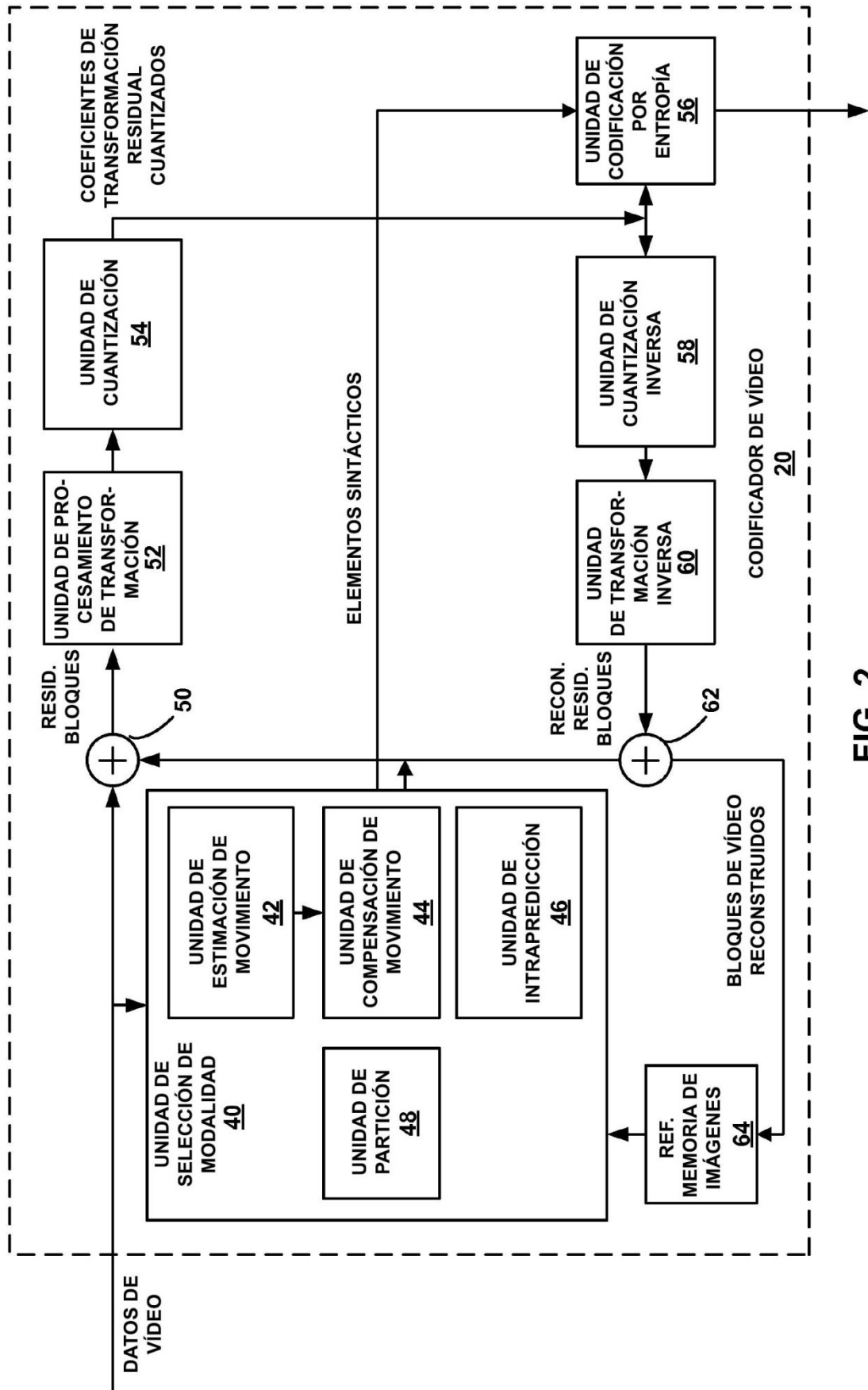


FIG. 2

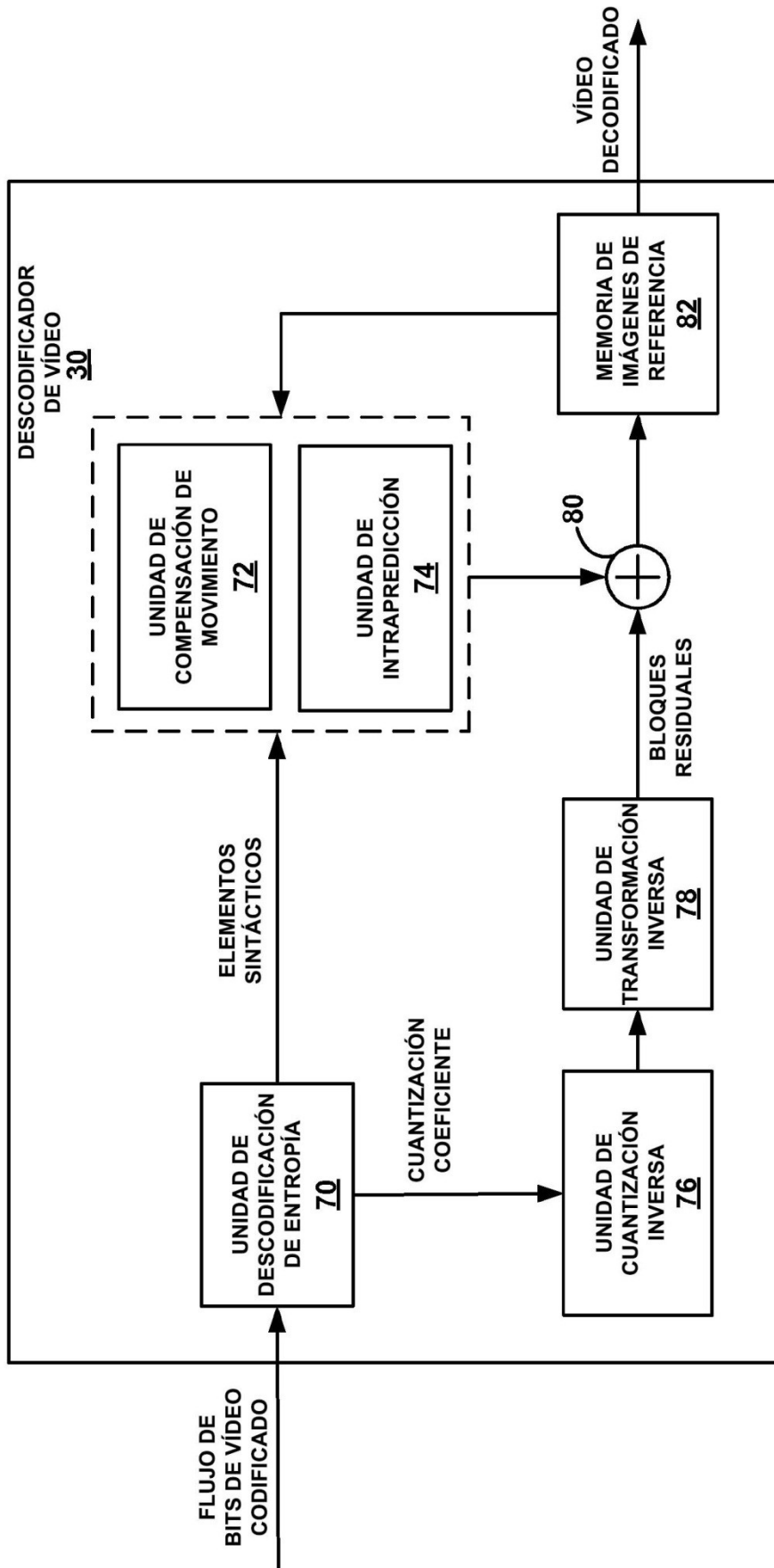
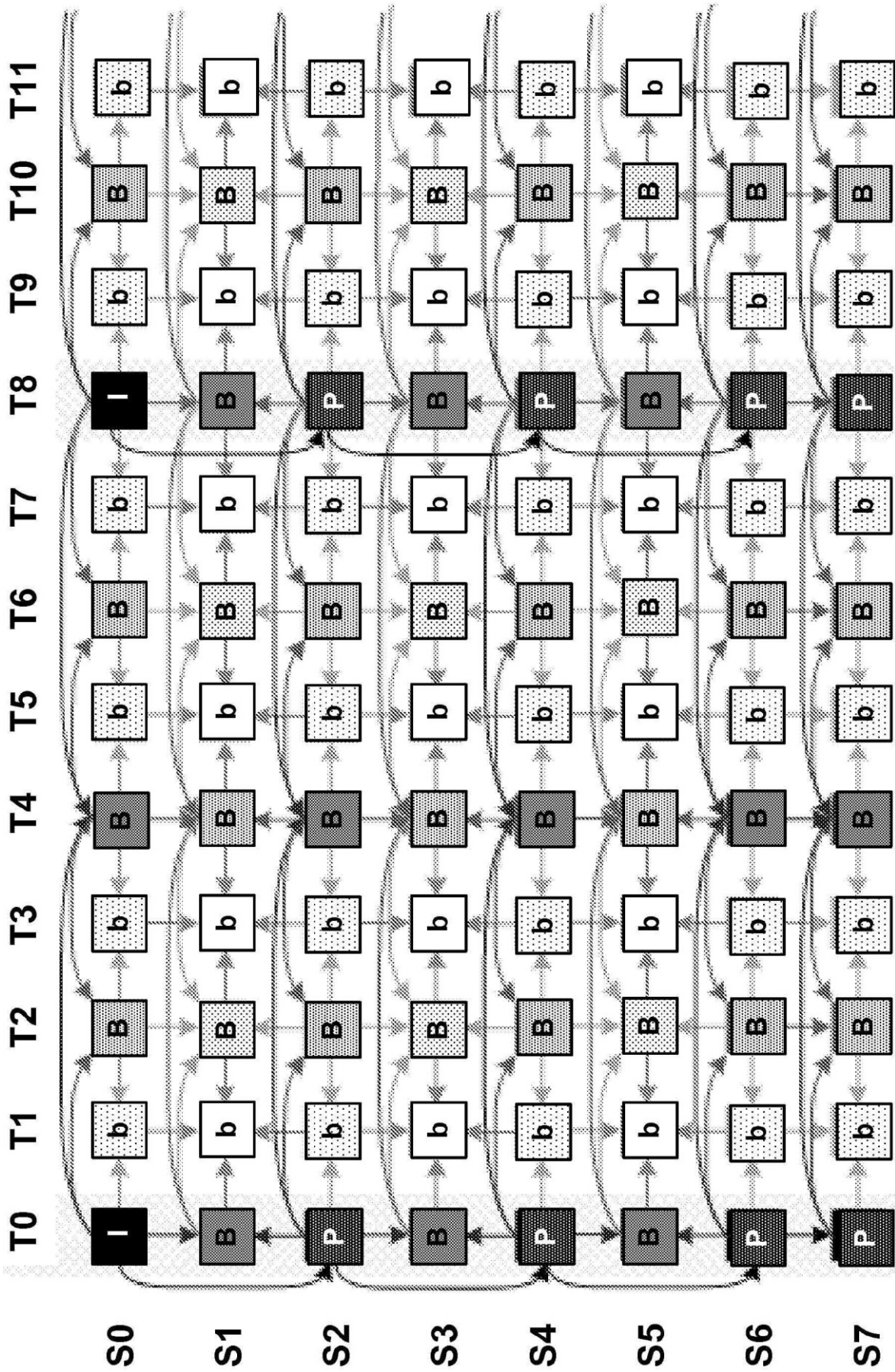


FIG. 3



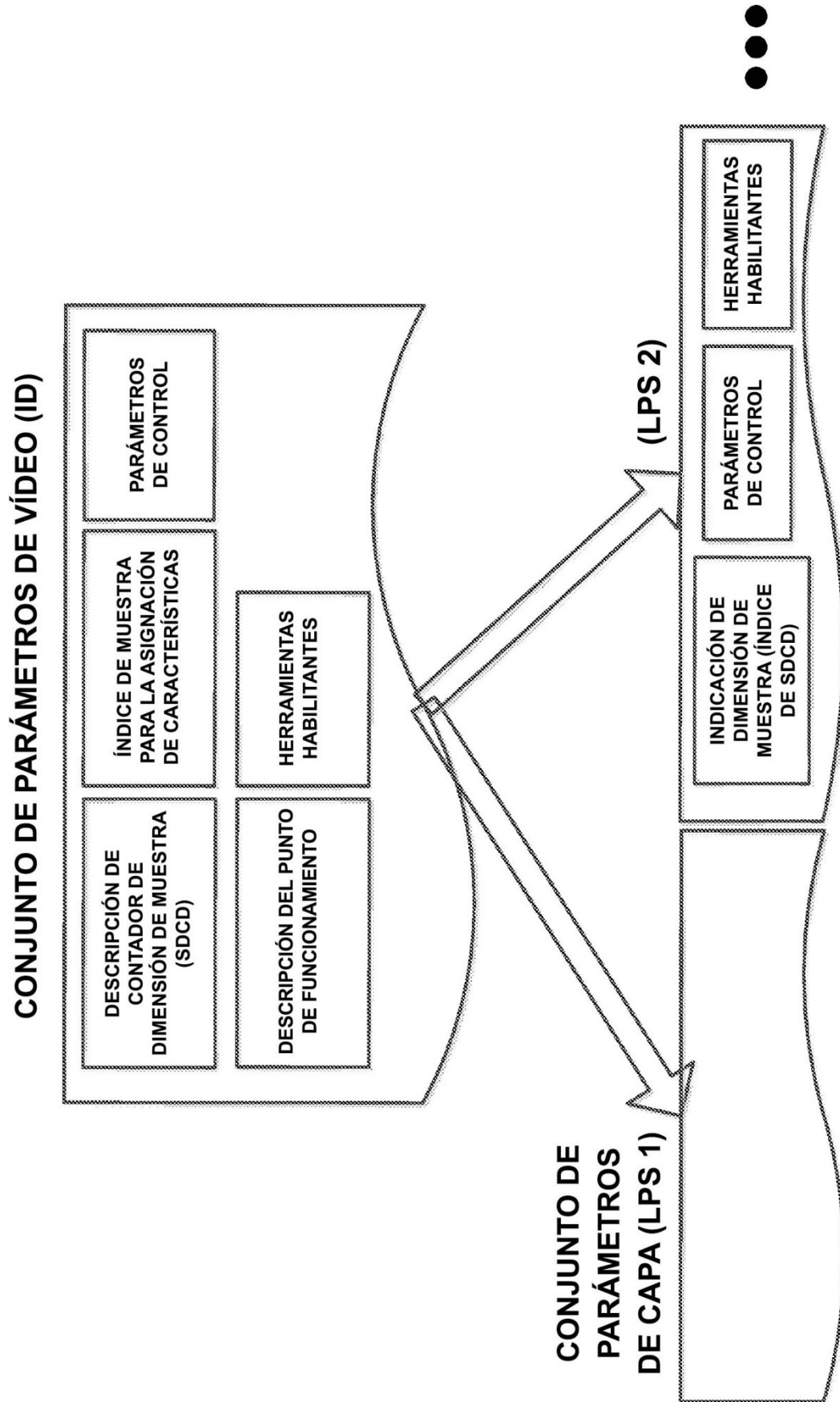


FIG. 5

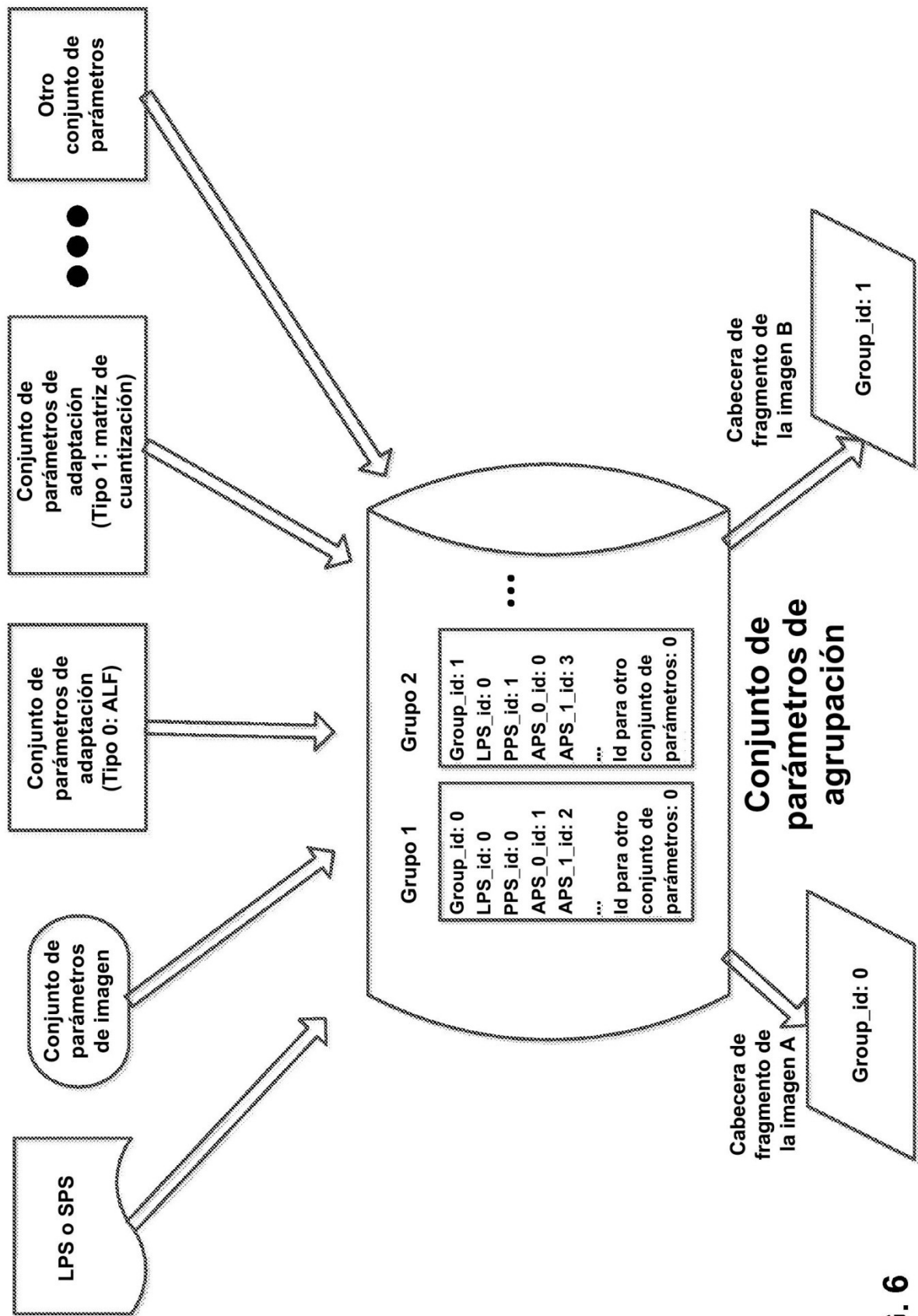


FIG. 6

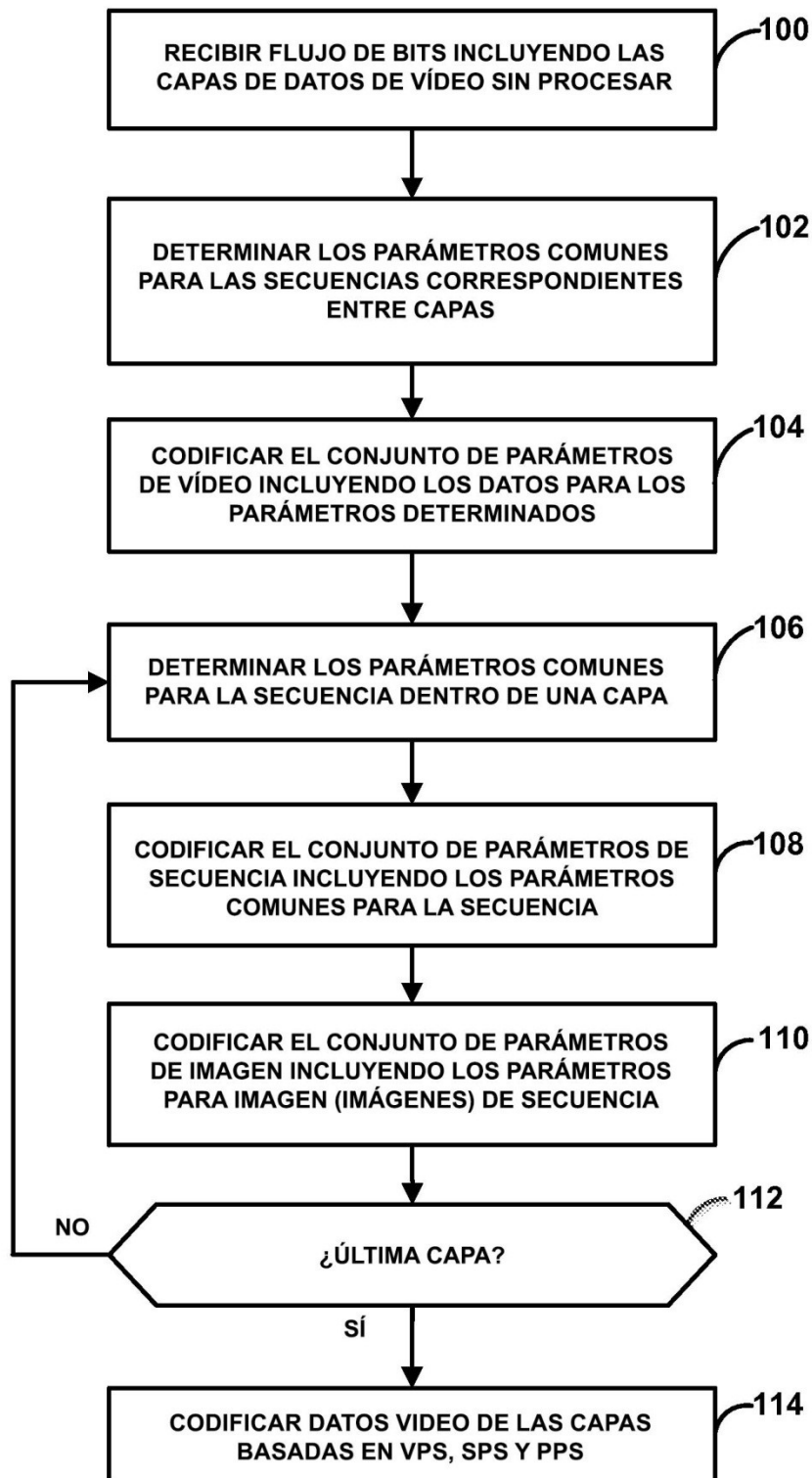


FIG. 7

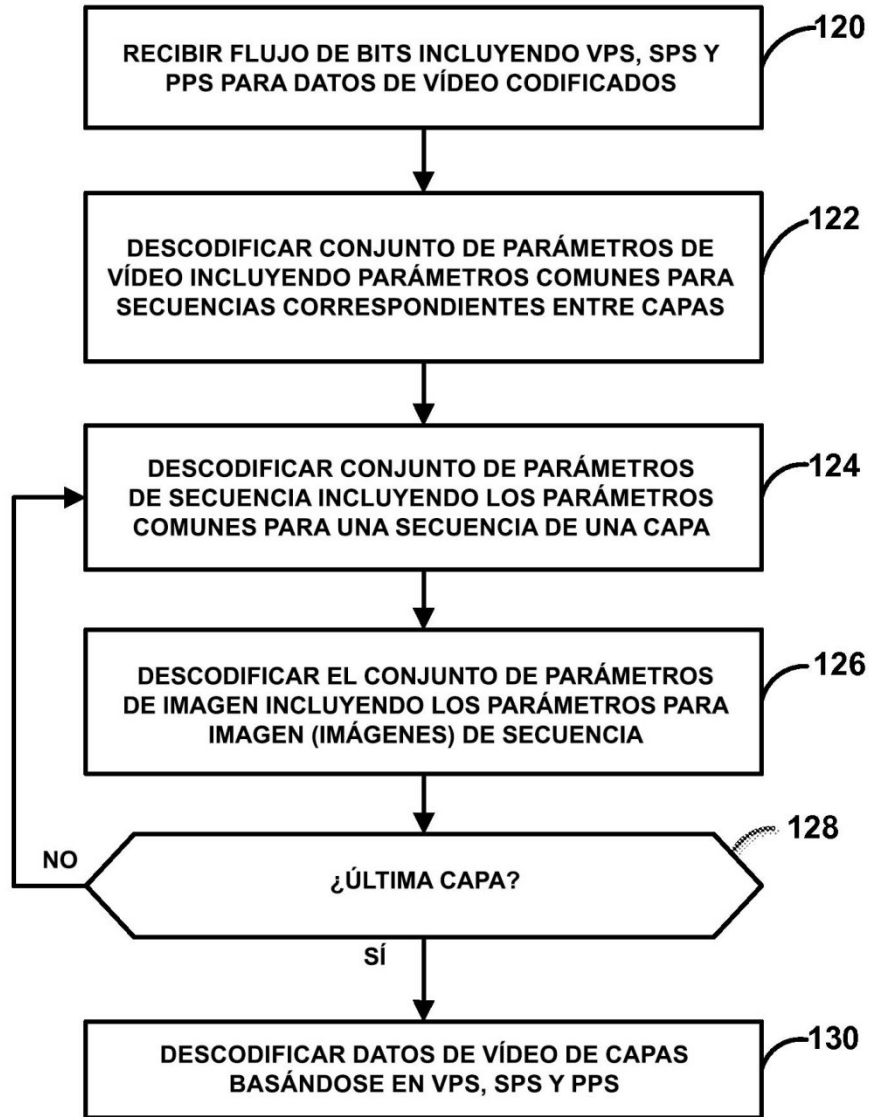


FIG. 8

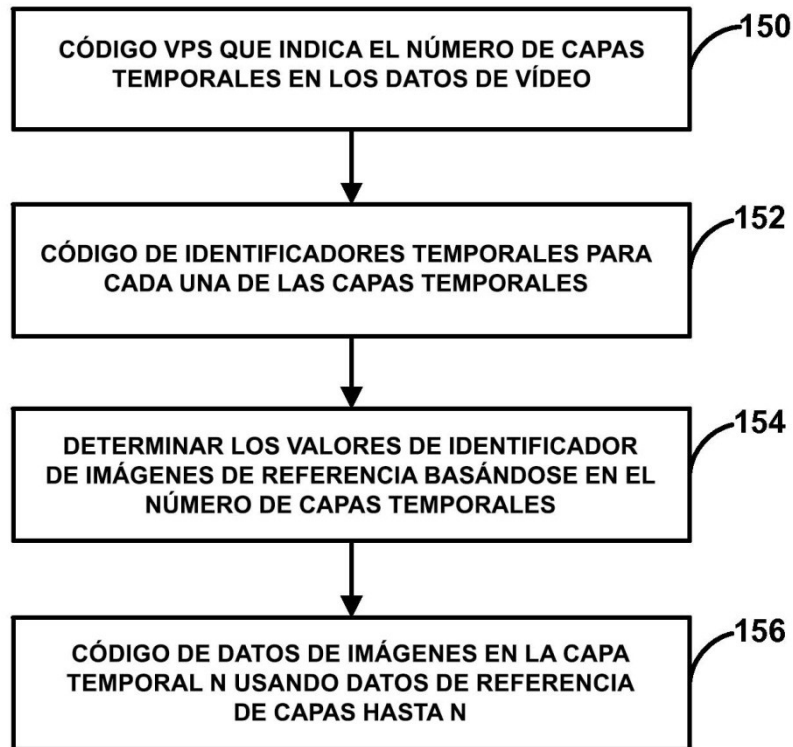


FIG. 9

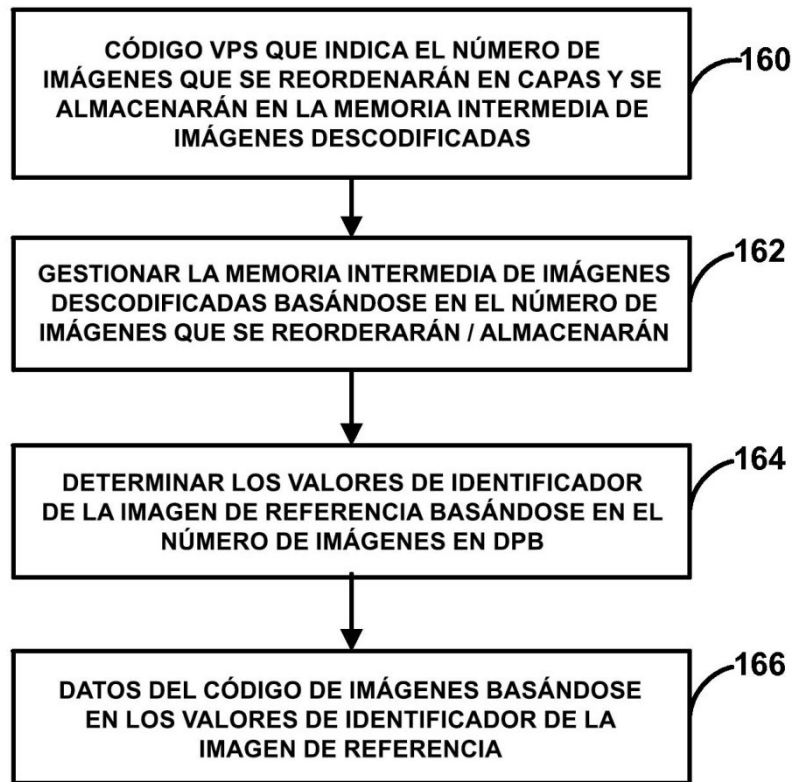


FIG. 10

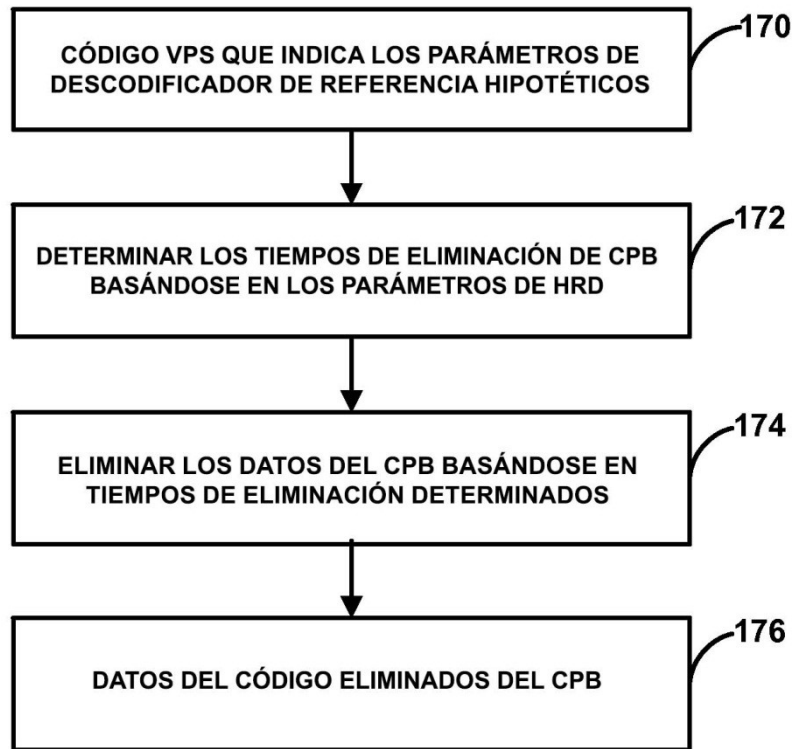


FIG. 11

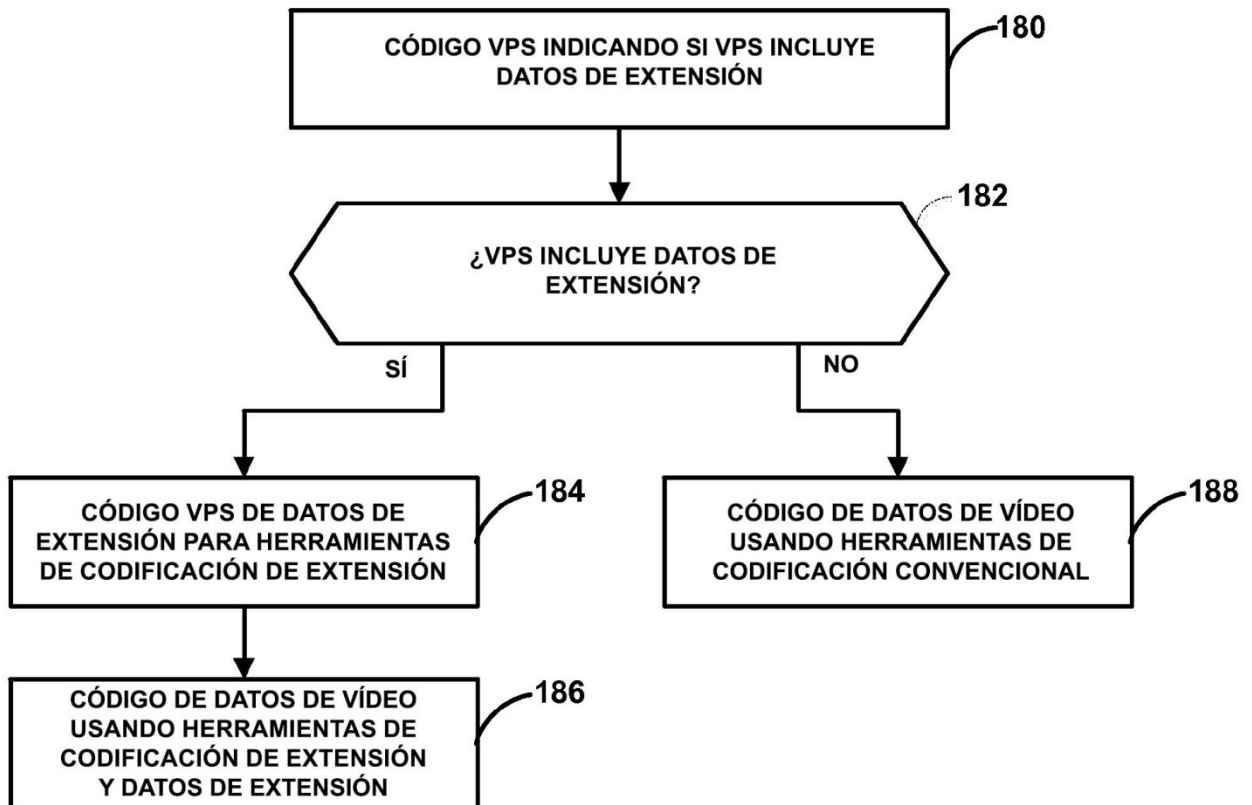


FIG. 12