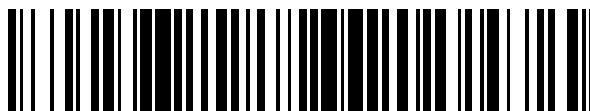


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 711 892**

51 Int. Cl.:

H04N 21/2343 (2011.01)

H04N 21/235 (2011.01)

H04N 21/435 (2011.01)

H04N 19/70 (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **11.02.2016 PCT/US2016/017499**

87 Fecha y número de publicación internacional: **18.08.2016 WO16130771**

96 Fecha de presentación y número de la solicitud europea: **11.02.2016 E 16712105 (2)**

97 Fecha y número de publicación de la concesión europea: **28.11.2018 EP 3257253**

54 Título: **Señalización de puntos de operación para el transporte de extensiones de la HEVC**

30 Prioridad:

11.02.2015 US 201562115089 P
10.02.2016 US 201615040418

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
08.05.2019

73 Titular/es:

QUALCOMM INCORPORATED (100.0%)
5775 Morehouse Drive
San Diego, CA 92121-1714, US

72 Inventor/es:

HENDRY, FNU;
CHEN, YING y
WANG, YE-KUI

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 711 892 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín Europeo de Patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre Concesión de Patentes Europeas).

DESCRIPCIÓN

Señalización de puntos de operación para el transporte de extensiones de la HEVC

5 **CAMPO TÉCNICO**

[0001] Esta divulgación se refiere a la codificación de vídeo y, más particularmente, al transporte de datos de vídeo codificados.

10 **ANTECEDENTES**

[0002] Las capacidades del vídeo digital pueden incorporarse en una amplia gama de dispositivos, incluyendo televisores digitales, sistemas de difusión directa digital, 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 por flujo 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 de Vídeo Avanzada (AVC), la norma de Codificación de Vídeo de Alta Eficacia (HEVC) y las extensiones de dichas normas. 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.

[0003] Las técnicas de codificación de vídeo incluyen la predicción espacial (intraimagen) 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 se codifican utilizando predicción espacial con respecto a muestras de referencia en bloques contiguos en 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 vecinos en 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.

[0004] 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 a 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, que a continuación se pueden cuantizar. Los coeficientes de transformación cuantizados, dispuestos inicialmente en una formación bidimensional, pueden examinarse con el fin de generar un vector unidimensional de coeficientes de transformación, y puede aplicarse codificación por entropía para lograr aún más compresión.

[0005] El documento "Proposed Study Text of ISO/IEC 13818-1:2013/PDAM7 [Texto de estudio propuesto de ISO / IEC 13818-1: 2013 / PDAM7]", de Grüneberg et al, describe las mejoras técnicas propuestas relacionadas con el transporte de datos de vídeo de la HEVC en capas.

50 **SUMARIO**

[0006] El alcance de la protección se define en las reivindicaciones independientes, a las que ahora se debe hacer referencia. Las características optativas se incluyen en las reivindicaciones dependientes.

[0007] En general, esta divulgación describe técnicas que pueden mejorar el diseño del descriptor del punto de operación de HEVC en el flujo de transporte (TS) de MPEG-2 (Grupo de Expertos en Imágenes en Movimiento) para el transporte. El transporte de datos de vídeo codificados también puede denominarse porte de datos de vídeo codificados. Las técnicas de esta divulgación pueden usarse para el transporte de datos de vídeo codificados para una extensión de una norma de codificación de vídeo, por ejemplo, una extensión de la norma de codificación de vídeo de alta eficacia (HEVC). Dichas extensiones pueden incluir extensiones de vista múltiple (por ejemplo, MV-HEVC), extensiones ajustables a escala (por ejemplo, SHVC) y extensiones en tres dimensiones (por ejemplo, 3D-HEVC). Sin embargo, debería entenderse que las técnicas de esta divulgación pueden usarse con otros flujos de transporte y / u otras técnicas de compresión de vídeo.

[0008] En un aspecto, esta divulgación describe un procedimiento para procesar datos de vídeo, comprendiendo el procedimiento: recibir un flujo de transporte que incluye un primer descriptor, un segundo descriptor y una pluralidad

de flujos elementales, siendo el primer descriptor un descriptor para un punto de operación de los datos de vídeo, siendo el segundo descriptor uno entre: un descriptor jerárquico o un descriptor de extensión jerárquica; decodificar un primer elemento sintáctico y un segundo elemento sintáctico en el primer descriptor, en donde: una lista de flujos elementales es una lista de flujos elementales del flujo de transporte que forman parte del punto de operación, el segundo descriptor tiene un valor de índice de capa jerárquica igual a un valor del segundo elemento sintáctico, un primer valor del primer elemento sintáctico especifica que: un flujo elemental indicado por el segundo elemento sintáctico, cuando no esté presente en la lista de flujos elementales, se agregará a la lista de flujos elementales, siendo el flujo elemental indicado por el segundo elemento sintáctico uno entre la pluralidad de flujos elementales, y un flujo elemental, indicado por un índice de capa incrustado en el segundo descriptor, cuando no esté presente en la lista de flujos elementales, se agregará a la lista de flujos elementales, siendo el flujo elemental indicado por el índice de capa incrustado en el segundo descriptor uno entre la pluralidad de flujos elementales; y, en respuesta a la determinación de que el primer elemento sintáctico tiene un segundo valor diferente al primer valor, agregar el flujo elemental indicado por el segundo elemento sintáctico, cuando no esté presente en la lista de flujos elementales, en la lista de flujos elementales, pero no agregar el flujo elemental, indicado por el índice de capa incrustado en el segundo descriptor, en la lista de flujos elementales.

[0009] En otro aspecto, esta divulgación describe un procedimiento para procesar datos de vídeo, comprendiendo el procedimiento: generar un primer descriptor, siendo el primer descriptor un descriptor de un punto de operación de los datos de vídeo; generar un segundo descriptor, siendo el segundo descriptor uno entre: un descriptor jerárquico o un descriptor de extensión jerárquica; en donde generar el primer descriptor comprende: incluir un primer elemento sintáctico y un segundo elemento sintáctico en el primer descriptor, en donde: una lista de flujos elementales es una lista de flujos elementales que forman parte del punto de operación, el segundo descriptor tiene un índice de capa jerárquica igual a un valor del segundo elemento sintáctico, un primer valor del primer elemento sintáctico especifica que: un flujo elemental indicado por el segundo elemento sintáctico, cuando no esté presente en la lista de flujos elementales, se agregará a la lista de flujos elementales, y un flujo elemental, indicado por un índice de capa incrustado en el segundo descriptor, cuando no esté presente en la lista de flujos elementales, se agregará a la lista de flujos elementales, y un segundo valor del primer elemento sintáctico especifica que el flujo elemental indicado por el segundo elemento sintáctico, cuando no esté presente en la lista de flujos elementales, se agregará a la lista de flujos elementales, pero no el flujo elemental indicado por el índice de capa incrustado en el segundo descriptor; e incluir el primer descriptor y el segundo descriptor en un flujo de transporte.

[0010] En otro aspecto, esta divulgación describe un dispositivo para procesar datos de vídeo, comprendiendo el dispositivo: una interfaz de entrada configurada para recibir información, incluyendo la información un flujo de transporte que incluye un primer descriptor, un segundo descriptor y una pluralidad de flujos elementales, siendo el primer descriptor un descriptor para un punto de operación de los datos de vídeo, siendo el segundo descriptor uno entre: un descriptor jerárquico o un descriptor de extensión jerárquica; y uno o más procesadores configurados para: decodificar un primer elemento sintáctico y un segundo elemento sintáctico en el primer descriptor, en donde: una lista de flujos elementales es una lista de flujos elementales del flujo de transporte que forman parte del punto de operación, el segundo descriptor tiene un valor de índice de capa jerárquica igual a un valor del segundo elemento sintáctico, un primer valor del primer elemento sintáctico especifica que: un flujo elemental indicado por el segundo elemento sintáctico, cuando no esté presente en la lista de flujos elementales, se agregará a la lista de flujos elementales, siendo el flujo elemental indicado por el segundo elemento sintáctico uno entre la pluralidad de flujos elementales, y un flujo elemental indicado por un índice de capa incrustado en el segundo descriptor, cuando no esté presente en la lista de flujos elementales, se agregará a la lista de flujos elementales, siendo el flujo elemental indicado por el índice de capa incrustado en el segundo descriptor uno entre la pluralidad de flujos elementales; y, en respuesta a la determinación de que el primer elemento sintáctico tiene un segundo valor diferente al primer valor, agregar el flujo elemental indicado por el segundo elemento sintáctico, cuando no esté presente en la lista de flujos elementales, en la lista de flujos elementales, pero no agregar el flujo elemental, indicado por el índice de capa incrustado en el segundo descriptor, en la lista de flujos elementales.

[0011] En otro aspecto, esta divulgación describe un dispositivo para procesar datos de vídeo, comprendiendo el dispositivo: uno o más procesadores configurados para: generar un primer descriptor, siendo el primer descriptor un descriptor de un punto de operación de los datos de vídeo; generar un segundo descriptor, siendo el segundo descriptor uno entre: un descriptor jerárquico o un descriptor de extensión jerárquica; en el que los uno o más procesadores están configurados de tal manera que, como parte de la generación del primer descriptor, los uno o más procesadores: incluyen un primer elemento sintáctico y un segundo elemento sintáctico en el primer descriptor, en donde: una lista de flujos elementales es una lista de flujos elementales que forman parte del punto de operación, el segundo descriptor tiene un valor de índice de capa jerárquica igual a un valor del segundo elemento sintáctico, un primer valor del primer elemento sintáctico especifica que: un flujo elemental indicado por el segundo elemento sintáctico, cuando no esté presente en la lista de flujos elementales, se agregará a la lista de flujos elementales, y un flujo elemental indicado por un índice de capa incrustado en el segundo descriptor, cuando no esté presente en la lista de flujos elementales, se agregará a la lista de flujos elementales, y un segundo valor del primer elemento sintáctico especifica que el flujo elemental indicado por el segundo elemento sintáctico, cuando no esté presente en la lista de flujos elementales, se agregará a la lista de flujos elementales, pero no el flujo elemental indicado por el índice de capa incrustado en el segundo descriptor; e incluir el primer descriptor y el segundo descriptor en un flujo de transporte; y una interfaz de salida configurada para generar una versión codificada de los datos de vídeo.

5 **[0012]** En otro aspecto, esta divulgación describe un dispositivo para procesar datos de vídeo, comprendiendo el dispositivo medios para recibir un flujo de transporte que incluye un primer descriptor, un segundo descriptor y una pluralidad de flujos elementales, siendo el primer descriptor un descriptor para un punto de operación de los datos de vídeo, siendo el segundo descriptor uno entre: un descriptor jerárquico o un descriptor de extensión jerárquica; medios para decodificar un primer elemento sintáctico y un segundo elemento sintáctico en el primer descriptor, en donde: una lista de flujos elementales es una lista de flujos elementales del flujo de transporte que forman parte del punto de operación, el segundo descriptor tiene un valor de índice de capa jerárquica igual a un valor del segundo elemento sintáctico, un primer valor del primer elemento sintáctico especifica que: un flujo elemental indicado por el segundo elemento sintáctico, cuando no esté presente en la lista de flujos elementales, se agregará a la lista de flujos elementales, siendo el flujo elemental indicado por el segundo elemento sintáctico uno entre la pluralidad de flujos elementales, y un flujo elemental indicado por un índice de capa incrustado en el segundo descriptor, cuando no esté presente en la lista de flujos elementales, se agregará a la lista de flujos elementales, siendo el flujo elemental indicado por el índice de capa incrustado en el segundo descriptor uno entre la pluralidad de flujos elementales, y medios para agregar, en respuesta a la determinación de que el primer elemento sintáctico tiene un segundo valor diferente al primer valor, el flujo elemental indicado por el segundo elemento sintáctico, cuando no está presente en la lista de flujos elementales, en la lista de flujos elementales, pero sin agregar el flujo elemental, indicado por el índice de capa incrustado en el segundo descriptor, en la lista de flujos elementales.

20 **[0013]** En otro aspecto, esta divulgación describe un dispositivo para procesar datos de vídeo, comprendiendo el dispositivo medios para generar un primer descriptor, siendo el primer descriptor un descriptor de un punto de operación de los datos de vídeo; medios para generar un segundo descriptor, siendo el segundo descriptor uno entre: un descriptor jerárquico o un descriptor de extensión jerárquica; en donde los medios para generar el primer descriptor comprenden: medios para incluir un primer elemento sintáctico y un segundo elemento sintáctico en el primer descriptor, en donde: una lista de flujos elementales es una lista de flujos elementales que forman parte del punto de operación, el segundo descriptor tiene un valor de índice de capa jerárquica igual a un valor del segundo elemento sintáctico, un primer valor del primer elemento sintáctico especifica que: un flujo elemental indicado por el segundo elemento sintáctico, cuando no esté presente en la lista de flujos elementales, se agregará a la lista de flujos elementales, y un flujo elemental indicado por un índice de capa incrustado en el segundo descriptor, cuando no esté presente en la lista de flujos elementales, se agregará a la lista de flujos elementales, y un segundo valor del primer elemento sintáctico especifica que el flujo elemental indicado por el segundo elemento sintáctico, cuando no esté presente en la lista de flujos elementales, se agregará a la lista de flujos elementales, pero no el flujo elemental indicado por el índice de capa incrustado en el segundo descriptor; y medios para incluir el primer descriptor y el segundo descriptor en un flujo de transporte.

35 **[0014]** En otro aspecto más, esta divulgación describe un medio de almacenamiento de datos legible por ordenador que comprende instrucciones almacenadas en el mismo que, cuando se ejecutan, hacen que un dispositivo para procesar datos de vídeo realice los procedimientos de los párrafos anteriores y descritos en esta divulgación.

40 **[0015]** 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 de los dibujos, y a partir de las reivindicaciones.

45 **BREVE DESCRIPCIÓN DE LOS DIBUJOS**

45 **[0016]**

50 La figura 1 es un diagrama de bloques que ilustra un sistema ejemplar de codificación y decodificación de vídeo que puede utilizar técnicas para transportar datos de vídeo codificados de acuerdo con extensiones de una norma de codificación de vídeo.

55 La figura 2 es un diagrama de bloques que ilustra un ejemplo de un codificador de vídeo que puede implementar técnicas para transportar datos de vídeo codificados de acuerdo con las extensiones de la norma de codificación de vídeo.

La figura 3 es un diagrama de bloques que ilustra un ejemplo de un decodificador de vídeo que puede implementar técnicas para transportar datos de vídeo codificados de acuerdo con las extensiones de la norma de codificación de vídeo.

60 La figura 4 es un diagrama de flujo que ilustra una operación ejemplar de un dispositivo que genera un flujo de transporte, de acuerdo con una técnica de esta divulgación.

65 La figura 5 es un diagrama de flujo que ilustra una operación ejemplar de un dispositivo que recibe un flujo de transporte, de acuerdo con una técnica de esta divulgación.

La figura 6 es un diagrama de flujo que ilustra una operación ejemplar de un dispositivo para generar un flujo de transporte de acuerdo con una técnica ejemplar de esta divulgación.

5 La figura 7 es un diagrama de flujo que ilustra una operación ejemplar de un dispositivo que recibe un flujo de transporte de acuerdo con una técnica ejemplar de esta divulgación.

DESCRIPCIÓN DETALLADA

10 **[0017]** En general, esta divulgación describe técnicas relacionadas con datos del nivel de Sistemas del grupo de expertos en imágenes en movimiento (MPEG)-2 para datos de medios. La especificación de sistemas de MPEG-2 generalmente describe cómo dos o más flujos de datos se multiplexan entre sí para formar un solo flujo de datos. Esta divulgación describe técnicas relacionadas con los datos de los sistemas de MPEG-2 para datos de vídeo de múltiples capas. Por ejemplo, esta divulgación describe cambios que potencialmente pueden mejorar el diseño del descriptor de punto de operación de la Codificación de vídeo de alta eficacia (HEVC) en flujos de transporte (TS) del MPEG-2 para el transporte de extensiones de la HEVC. Sin embargo, las técnicas de esta divulgación no están necesariamente limitadas al TS del MPEG-2 o a la HEVC.

20 **[0018]** Solo para facilitar la descripción, las técnicas de esta divulgación se describen generalmente para el porte (por ejemplo, el transporte) de datos de vídeo codificados de acuerdo con una extensión para una norma de codificación de vídeo (por ejemplo, una extensión a la HEVC, también conocida como ITU-T H.265). Dichas extensiones pueden incluir extensiones de vista múltiple, tridimensionales y / o ajustables a escala. Por lo tanto, las técnicas de esta divulgación pueden aplicarse a la HEVC de vista múltiple (MV-HEVC), la HEVC tridimensional (3D-HEVC) y la HEVC ajustable a escala (SHVC).

25 **[0019]** Los datos de vídeo de múltiples capas, por ejemplo, los datos de vídeo de múltiples vistas y / o los datos de vídeo con múltiples capas ajustables a escala, pueden incluir puntos de operación designados. En general, un punto de operación describe un subconjunto de capas (por ejemplo, vistas) de un conjunto completo de capas de datos de vídeo de múltiples capas. El punto de operación también puede identificar las capas de salida de destino, es decir, las capas para las cuales se han de emitir los datos. En algunos casos, los datos de una capa pueden incluirse en un punto de operación solo para su uso como capa de referencia y, por lo tanto, dicha capa no se considerará una capa de salida de destino. Una capa de referencia puede ser una capa utilizada como referencia por otra capa para la codificación y decodificación.

35 **[0020]** Los puntos de operación de la HEVC generalmente se señalizan en un descriptor de punto de operación de la HEVC especificando referencias a flujos elementales, como se describe con más detalle a continuación. Sin embargo, algunos diseños del descriptor de punto de operación de la HEVC permiten una lista de flujos elementales para que un punto de operación de destino incluya una capa o un flujo elemental dos veces. Además, en algunos casos, no todas las capas que deben decodificarse están señalizadas en algunos diseños del descriptor del punto de operación de la HEVC porque la información para dichas capas está disponible en otros lugares. En algunos ejemplos, debido a que una capa particular no está señalizada, ciertos elementos sintácticos asociados a la capa pueden no estar disponibles para el decodificador de vídeo.

45 **[0021]** Las técnicas descritas en esta divulgación pueden abordar una o más de las cuestiones descritas anteriormente. Sin embargo, no es una necesidad que las técnicas aborden las cuestiones descritas anteriormente. Por ejemplo, la divulgación describe ejemplos que pueden garantizar que un flujo, o una capa, elemental no se incluya varias veces en la lista de flujos elementales del punto de operación. La divulgación también describe ejemplos de inclusión de información en el flujo de bits que recibe un decodificador de vídeo que incluye elementos sintácticos asociados a una capa, incluso si esa capa no se señala explícitamente como parte de la lista de flujos elementales del punto de operación.

50 **[0022]** La figura 1 es un diagrama de bloques que ilustra un sistema ejemplar de codificación y decodificación de vídeo 10 que puede utilizar técnicas para transportar datos de vídeo codificados de acuerdo con extensiones de una norma de codificación de vídeo. Como se muestra en la figura 1, el sistema 10 incluye un dispositivo de origen 12 que proporciona datos de vídeo codificados, a ser decodificados en un momento posterior por un dispositivo de destino 14. En particular, el dispositivo de origen 12 proporciona los datos de vídeo al dispositivo de destino 14 mediante un medio legible por ordenador 16. El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender cualquiera entre una amplia gama de dispositivos, incluyendo ordenadores de escritorio, ordenadores plegables (es decir, portátiles), ordenadores de tableta, decodificadores, equipos telefónicos de mano tales como los denominados teléfonos "inteligentes", tabletas, televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivo de transmisión por flujo de vídeo o similares. En algunos ejemplos, el dispositivo de origen 12 y el dispositivo de destino 14 pueden estar equipados para la comunicación inalámbrica. Por lo tanto, en algunos ejemplos, el dispositivo de origen 12 y el dispositivo de destino 14 son dispositivos de comunicación inalámbrica.

65 **[0023]** En el ejemplo de la figura 1, el dispositivo de origen 12 incluye un origen de vídeo 18, un codificador de vídeo 20, un multiplexador 21 y una interfaz de salida 22. En algunos ejemplos, la interfaz de salida 22 incluye un

modulador/demodulador (módem) y/o un transmisor. El origen de vídeo 18 puede incluir un origen tal como un dispositivo de captura de vídeo, por ejemplo, una videocámara, un archivo de vídeo que contiene vídeo previamente capturado, una interfaz de alimentación de vídeo para recibir vídeo desde un proveedor de contenido de vídeo y/o un sistema de gráficos de ordenador para generar datos de gráficos de ordenador como el vídeo de origen, o una combinación de dichos orígenes. Sin embargo, las técnicas descritas en esta divulgación pueden ser aplicables a la codificación de vídeo en general, y se pueden aplicar a aplicaciones inalámbricas y/o cableadas.

[0024] El codificador de vídeo 20 puede codificar datos de vídeo capturados, precapturados o generados por ordenador. El dispositivo de origen 12 puede transmitir los datos de vídeo codificados directamente al dispositivo de destino 14 mediante la interfaz de salida 22 del dispositivo de origen 12. Los datos de vídeo codificados pueden almacenarse también (o de forma alternativa) en un dispositivo de almacenamiento para un acceso posterior por parte del dispositivo de destino 14 u otros dispositivos, para su decodificación y/o reproducción. La información de vídeo codificada puede entonces ser emitida por la interfaz de salida 22 hacia un medio legible por ordenador 16.

[0025] El dispositivo de destino 14 incluye una interfaz de entrada 28, un demultiplexador 29, un decodificador de vídeo 30 y un dispositivo de visualización 32. En algunos casos, la interfaz de entrada 28 puede incluir un receptor y/o un módem. La interfaz de entrada 28 del dispositivo de destino 14 recibe los datos de vídeo codificados por un enlace 16. Los datos de vídeo codificados, comunicados por el enlace 16, o proporcionados en el dispositivo de almacenamiento 33, pueden incluir una diversidad de elementos sintácticos generados por el codificador de vídeo 20, para su uso por un decodificador de vídeo, tal como el decodificador de vídeo 30, en la decodificación de los datos de vídeo. Dichos elementos sintácticos se pueden incluir con los datos de vídeo codificados, transmitidos en un medio de comunicación, almacenar en un medio de almacenamiento o almacenar en un servidor de ficheros.

[0026] El dispositivo de destino 14 puede recibir los datos de vídeo codificados, a decodificar 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 de origen 12 al dispositivo de destino 14. En un ejemplo, el medio legible por ordenador 16 puede comprender un medio de comunicación que habilita el dispositivo de origen 12 para transmitir datos de vídeo codificados directamente al dispositivo de destino 14 en tiempo real. Los datos de vídeo codificados se pueden modular de acuerdo con una norma de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitir al dispositivo de destino 14. El medio de comunicación puede comprender cualquier medio de comunicación, inalámbrica o cableada, tal como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión física. El medio de comunicación puede formar parte de una red basada en paquetes, tal como una red de área local, una red de área amplia o una red global tal como Internet. El medio de comunicación puede incluir encaminadores, conmutadores, estaciones base o cualquier otro equipo que pueda ser útil para facilitar la comunicación desde el dispositivo de origen 12 hasta el dispositivo de destino 14.

[0027] En algunos ejemplos, pueden emitirse datos codificados desde la interfaz de salida 22 hasta un dispositivo de almacenamiento. Por ejemplo, el dispositivo de origen 12 puede incluir un dispositivo de almacenamiento configurado para almacenar los datos de vídeo codificados. En algunos ejemplos, se puede acceder a los datos codificados desde el dispositivo de almacenamiento mediante la interfaz de entrada 28. 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 duro, 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 de origen 12. El dispositivo de destino 14 puede acceder a datos de vídeo almacenados desde el dispositivo de almacenamiento mediante transmisión por 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 ejemplares incluyen servidores de la Red (por ejemplo, para una sede de la Red), servidores del FTP, dispositivos de almacenamiento conectados a la red (NAS), o unidades de disco locales u otros servidores configurados para proporcionar ficheros. El dispositivo de destino 14 puede acceder a los datos de vídeo codificados a través de cualquier conexión de datos estándar, incluyendo una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión de Wi-Fi), una conexión por cable (por ejemplo, un módem DSL, un módem por cable), o una combinación de ambos que sea adecuada para acceder a datos de vídeo codificados 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.

[0028] El medio legible por ordenador 16 puede incluir medios transitorios, tales como una difusión 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 otros medios legibles por ordenador. En algunos ejemplos, un servidor de red (no mostrado) puede recibir datos de vídeo codificados desde el dispositivo de origen 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 utilidad de producción de un medio, tal como una utilidad de grabación de discos, puede recibir datos de vídeo codificados desde el dispositivo de origen 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 diversas formas, en varios ejemplos.

5 **[0029]** Las técnicas de esta divulgación no están limitadas necesariamente a aplicaciones o configuraciones inalámbricas. Las técnicas pueden aplicarse a la codificación de vídeo como soporte de cualquiera entre una diversidad de aplicaciones de multimedios, tales como difusiones de televisión por el aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones por flujo de vídeo por Internet, tales como el flujo de transmisión adaptativo dinámico sobre HTTP (DASH), el vídeo digital que se codifica en un medio de almacenamiento de datos, la decodificación de vídeo digital almacenado en un medio de almacenamiento de datos, u otras aplicaciones.
10 En algunos ejemplos, el sistema 10 se puede configurar para prestar soporte a la transmisión de vídeo unidireccional o bidireccional, para prestar soporte a aplicaciones tales como la transmisión por flujo de vídeo, la reproducción de vídeo, la difusión de vídeo y/o la videotelefonía.

15 **[0030]** En el ejemplo de la figura 1, de acuerdo con esta divulgación, el multiplexador 21 del dispositivo de origen 12 puede configurarse para aplicar las técnicas para transportar datos de vídeo codificados de acuerdo con las extensiones de una norma de codificación de vídeo, mientras que el demultiplexador 29 puede recibir dichos datos para procesar y remitir los datos de vídeo procesados a otro dispositivo o componente, por ejemplo, el decodificador de vídeo 30. En otros ejemplos, un dispositivo de origen y un dispositivo de destino pueden incluir otros componentes o disposiciones. Por ejemplo, el dispositivo de origen 12 puede recibir datos de vídeo desde un origen de vídeo externo 18, tal como una cámara externa. Del mismo modo, el dispositivo de destino 14 puede interactuar con un dispositivo de visualización externo, en lugar de incluir un dispositivo de visualización integrado.

25 **[0031]** El sistema ilustrado 10 de la figura 1 es simplemente un ejemplo. Las técnicas para el transporte de datos de vídeo, codificados de acuerdo con las extensiones de una norma de codificación 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 realizan mediante un dispositivo de codificación de vídeo, las técnicas también pueden realizarse mediante un codificador/decodificador de vídeo, denominado habitualmente «CÓDEC». El dispositivo de origen 12 y el dispositivo de destino 14 son simplemente ejemplos de dichos dispositivos de codificación en los que el dispositivo de origen 12 genera datos de vídeo codificados para su transmisión al dispositivo de destino 14. En algunos ejemplos, los dispositivos 12, 14 pueden funcionar de manera 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 prestar soporte a la transmisión de vídeo unidireccional o bidireccional entre los dispositivos de vídeo 12, 14, por ejemplo, para la transmisión de vídeo por flujo, la reproducción de vídeo, la difusión de vídeo o la videotelefonía.

35 **[0032]** 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 usado por el decodificador de vídeo 30, que incluye elementos sintácticos que describen características y/o procesamiento de bloques y de otras unidades codificadas.

40 **[0033]** El dispositivo de visualización 32 puede estar integrado con, o ser externo a, el dispositivo de destino 14. En algunos ejemplos, el dispositivo de destino 14 puede incluir un dispositivo de visualización integrado y también puede estar configurado para interactuar con un dispositivo de visualización externo. En otros ejemplos, el dispositivo de destino 14 puede ser un dispositivo de visualización. El dispositivo de visualización 32 muestra los datos de vídeo decodificados a un usuario, y puede comprender cualquiera entre una diversidad 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.

45 **[0034]** El codificador de vídeo 20, el multiplexador 21, el demultiplexador 29 y el decodificador de vídeo 30 pueden implementarse como cualquiera entre una variedad de circuitos adecuados de codificadores o decodificadores, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), formaciones de compuertas programables in situ (FPGA), lógica discreta, software, hardware, firmware o cualquier combinación de los mismos. Cuando las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio adecuado no transitorio legible por ordenador, y ejecutar las instrucciones en hardware utilizando uno o más procesadores para realizar las técnicas de esta divulgación. Cada uno entre el codificador de vídeo 20 y el decodificador de vídeo 30 se puede incluir en uno o más codificadores o decodificadores, cualquiera de los cuales se puede integrar como parte de un codificador/decodificador (CÓDEC) combinado en un dispositivo respectivo.

50 **[0035]** Esta divulgación puede referirse, en general, al codificador de vídeo 20 que "señaliza" o "transmite" cierta información a otro dispositivo, tal como el decodificador de vídeo 30. El término "señalizar" o "transmitir" puede referirse en general a la comunicación de elementos sintácticos y/u otros datos usados para decodificar los datos de vídeo comprimidos. Dicha comunicación puede producirse en tiempo real o casi real. De forma alternativa, dicha comunicación puede producirse durante un tramo de tiempo, tal como podría producirse cuando se almacenan elementos sintácticos en un medio de almacenamiento legible por ordenador en un flujo de bits codificado en el momento de la codificación, que entonces pueden ser recuperado por un dispositivo de decodificación en cualquier momento tras haber sido almacenado en este medio. Por lo tanto, aunque el decodificador de vídeo 30 puede

mencionarse como "recibiendo" cierta información, la recepción de información no necesariamente se produce en tiempo real o casi real, y puede recuperarse desde un medio en algún momento después del almacenamiento.

[0036] Además, aunque no se ilustra específicamente en la figura 1, algunas de las técnicas ejemplares descritas en esta divulgación pueden ser implementadas por algún medio externo, tal como un elemento de red consciente de los medios (MANE). El MANE puede recibir datos de vídeo desde el codificador de vídeo 20 y señalar un flujo de bits que recibe el decodificador de vídeo 30. En algunos ejemplos, el MANE puede recibir y procesar un flujo de transporte. En consecuencia, en ocasiones, cuando la divulgación describe la señalización, dicha señalización puede ser realizada por el codificador de vídeo 20 o por algún medio externo, tal como el MANE

[0037] El codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar de acuerdo con una norma de codificación de vídeo. De forma alternativa, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar de acuerdo con otras normas de propiedad o industriales, tales como la norma ITU-T H.264, alternativamente denominada MPEG-4, parte 10, Codificación Avanzada de Vídeo (AVC), o extensiones 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. La norma ITU-T H.264/MPEG-4 (AVC) fue formulada por el Grupo de Expertos en Codificación de Vídeo (VCEG) de la ITU-T junto con el Grupo de Expertos en Imágenes en Movimiento (MPEG) de la ISO/IEC, como el producto de una asociación colectiva conocida como Equipo Mixto de Vídeo (JVT, por sus siglas en inglés). En algunos aspectos, las técnicas descritas en esta divulgación pueden aplicarse a dispositivos que se ajustan en general a la norma H.264. La norma H.264 se describe en la recomendación H.264 de ITU-T, Codificación avanzada de vídeo para servicios audiovisuales genéricos, por el Grupo de estudio de la ITU-T, con fecha de marzo de 2005, que puede mencionarse en el presente documento como la norma H.264 o la especificación H.264, o la norma o especificación H.264/AVC. El Equipo mixto de vídeo (JVT) continúa trabajando en extensiones de la norma H.264/MPEG-4 AVC. Más en general, las normas de codificación de vídeo incluyen ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 o ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual e ITU-T H.264 (también conocida como ISO/IEC MPEG-4 AVC), incluyendo sus extensiones de codificación de vídeo ajustable a escala (SVC) y de codificación de vídeo de vista múltiple (MVC).

[0038] En otro ejemplo, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden codificar y decodificar datos de vídeo usando otra norma de codificación de vídeo, tal como la norma de Codificación de vídeo de alta eficacia (HEVC), también conocida como la norma de codificación de vídeo H.265. La HEVC ha sido finalizada por el Equipo Conjunto de Colaboración en Codificación de Vídeo (JCT-VC) del Grupo de Expertos en Codificación de Vídeo (VCEG) del ITU-T y el Grupo de Expertos en Imágenes en Movimiento (MPEG) de la ISO / IEC. La más reciente especificación preliminar de la HEVC se denomina HEVC WD en lo sucesivo.

[0039] Además, el JCT-3V también está desarrollando la extensión de vista múltiple para la HEVC, es decir, MV-HEVC. La ampliación ajustable a escala para la HEVC, llamada SHVC, también está siendo desarrollada por el JCT-VC. El documento más reciente que incluye la especificación para la HEVC, la extensión de rango de la HEVC, la SHVC y la MV-HEVC está disponible en http://phenix.int-evry.fr/jct/doc_end_user/documents/18_Sapporo/wg11/JCTVC-R1013-v6.zip.

[0040] En la HEVC y en otras normas de codificación de vídeo, una secuencia de vídeo incluye típicamente una serie de imágenes. Las imágenes también pueden denominarse "tramas". Para generar una representación codificada de una imagen, el codificador de vídeo 20 puede generar un conjunto de unidades arboladas de codificación (CTU). Cada una de las CTU puede ser un bloque arbolado de codificación de muestras de luminancia, dos correspondientes bloques arbolados de codificación de muestras de crominancia y estructuras sintácticas usadas para codificar las muestras de los bloques arbolados de codificación. Un bloque de árbol de codificación puede ser un bloque de muestras de tamaño NxN. Una CTU también puede denominarse "bloque arbolado" o "máxima unidad de codificación" (LCU).

[0041] Para generar una CTU codificada, el codificador de vídeo 20 puede dividir los bloques arbolados de codificación de una CTU en bloques de codificación, de ahí el nombre de "unidades arboladas de codificación". Una CU puede ser un bloque de codificación de muestras de luminancia y dos correspondientes bloques de codificación de muestras de crominancia de una imagen que tenga una formación de muestras de luminancia, una formación de muestras de Cb y una formación de muestras de Cr, y estructuras sintácticas usadas para codificar las muestras de los bloques de codificación. El codificador de vídeo 20 puede dividir un bloque de codificación de una CU en uno o más bloques de predicción. Un bloque de predicción puede ser un bloque rectangular (es decir, cuadrado o no cuadrado) de muestras en las que se aplique la misma predicción. Una unidad de predicción (PU) de una CU puede ser un bloque de predicción de muestras de luma, dos correspondientes bloques de predicción de muestras de croma de una imagen y estructuras sintácticas usadas para predecir las muestras de bloques de predicción. El codificador de vídeo 20 puede generar bloques predictivos de luma, Cb y Cr para bloques de predicción de luma, Cb y Cr de cada PU de la CU. El codificador de vídeo 20 puede usar intrapredicción o interpredicción para generar los bloques predictivos para una PU.

[0042] Después de que el codificador de vídeo 20 genera bloques predictivos para una o más PU de una CU, el codificador de vídeo 20 puede generar bloques residuales para la CU. Cada muestra en un bloque residual de la CU

puede indicar una diferencia entre una muestra en un bloque predictivo de una PU de la CU y una muestra correspondiente en un bloque de codificación de la CU. Además, el codificador de vídeo 20 puede descomponer los bloques residuales de una CU en uno o más bloques de transformación. Una unidad de transformación (TU) de una CU puede ser un bloque de transformación de muestras de luma, dos bloques de transformación correspondientes de muestras de croma y estructuras sintácticas usadas para transformar las muestras de bloques de transformación. El codificador de vídeo 20 puede aplicar una o más transformaciones a un bloque de transformación para generar un bloque de coeficientes para una TU. Después de generar un bloque de coeficientes, el codificador de vídeo 20 puede cuantizar el bloque de coeficientes. Después de que el codificador de vídeo 20 cuantice un bloque de coeficientes, el codificador de vídeo 20 puede codificar por entropía elementos sintácticos que indican los coeficientes de transformación cuantizados. Por ejemplo, el codificador de vídeo 20 puede realizar la Codificación Aritmética Binaria Adaptativa al Contexto (CABAC) en los elementos sintácticos que indiquen los coeficientes de transformación cuantizados. El codificador de vídeo 20 puede emitir los elementos sintácticos codificados por entropía en un flujo de bits. El flujo de bits también puede incluir elementos sintácticos que no estén codificados por entropía.

[0043] El flujo de bits puede incluir una secuencia de bits que forme una representación de imágenes codificadas y datos asociados. El flujo de bits puede comprender una secuencia de unidades de capas de abstracción de red (NAL). Cada una de las unidades de NAL incluye un encabezado de unidad de NAL y encapsula una carga útil de secuencias de octetos sin procesar (RBSP). Una RBSP puede ser una estructura sintáctica que contenga un número entero de octetos que se encapsule dentro de una unidad de NAL. En algunos casos, una RBSP incluye cero bits.

[0044] Diferentes tipos de unidades de NAL pueden encapsular diferentes tipos de las RBSP. Por ejemplo, un primer tipo de unidad de NAL puede encapsular una RBSP para un conjunto de parámetros de imagen (PPS), un segundo tipo de unidad de NAL puede encapsular una RBSP para un fragmento codificado, un tercer tipo de unidad de NAL puede encapsular una RBSP para información de mejora complementaria (SEI), y así sucesivamente. Las unidades de NAL que encapsulan las RBSP para datos de codificación de vídeo (a diferencia de las RBSP para conjuntos de parámetros y mensajes de SEI) pueden denominarse unidades de NAL de la capa de codificación de vídeo (VCL).

[0045] El decodificador de vídeo 30 puede recibir un flujo de bits generado por el codificador de vídeo 20. Además, el decodificador de vídeo 30 puede obtener elementos sintácticos del flujo de bits. Por ejemplo, el decodificador de vídeo 30 puede analizar sintácticamente el flujo de bits para decodificar elementos sintácticos a partir del flujo de bits. El decodificador de vídeo 30 puede reconstruir las imágenes de los datos de vídeo basándose, al menos en parte, en los elementos sintácticos obtenidos (por ejemplo, decodificados) a partir del flujo de bits. El proceso para reconstruir los datos de vídeo puede ser, en general, recíproco al proceso realizado por el codificador de vídeo 20. El decodificador de vídeo 30 puede reconstruir los bloques de codificación de la CU actual añadiendo las muestras de los bloques predictivos de muestra para las PU de la CU actual a las muestras correspondientes de los bloques de transformación de las TU de la CU actual. Mediante la reconstrucción de los bloques de codificación para cada CU de una imagen, el decodificador de vídeo 30 puede reconstruir la imagen.

[0046] En la codificación de vista múltiple, puede haber varias vistas de la misma escena desde diferentes puntos de vista. En el contexto de la codificación de vista múltiple, el término "unidad de acceso" se puede usar para referirse al conjunto de imágenes que corresponden a la misma instancia de tiempo. Por tanto, los datos de vídeo pueden conceptualizarse como una serie de unidades de acceso que se producen a lo largo del tiempo. Un "componente de vista" puede ser una representación codificada de una vista en una única unidad de acceso. En esta divulgación, una "vista" puede referirse a una secuencia de componentes de vistas asociadas al mismo identificador de vista. En algunos ejemplos, un componente de vista puede ser un componente de vista de textura (es decir, una imagen de textura) o un componente de vista de profundidad (es decir, una imagen de profundidad).

[0047] En la MV-HEVC y la SHVC, un codificador de vídeo puede generar un flujo de bits que comprende una serie de unidades de NAL. Las diferentes unidades de NAL del flujo de bits pueden estar asociadas a diferentes capas del flujo de bits. Una capa se puede definir como un conjunto de unidades de NAL de VCL y unidades asociadas de NAL no de VCL, que tienen el mismo identificador de capa. Una capa puede ser equivalente a una vista en la codificación de vídeo de vista múltiple. En la codificación de vídeo de vista múltiple, una capa puede contener todos los componentes de vista de la misma capa con diferentes casos en el tiempo. Cada componente de vista puede ser una imagen codificada de la escena de vídeo que pertenece a una vista específica en un caso específico en el tiempo. En algunos ejemplos de codificación de vídeo de vista múltiple o tridimensional, una capa puede contener todas las imágenes de profundidad codificadas de una vista específica o imágenes de textura codificadas de una vista específica. En otros ejemplos de codificación de vídeo tridimensional, una capa puede contener tanto componentes de vista de textura como componentes de vista de profundidad de una vista específica. De forma similar, en el contexto de la codificación de vídeo ajustable a escala, una capa corresponde habitualmente a imágenes codificadas que tengan características de vídeo diferentes a las imágenes codificadas en otras capas. Dichas características de vídeo habitualmente incluyen la resolución espacial y el nivel de calidad (por ejemplo, la razón entre señal y ruido). En la HEVC y en sus extensiones, la ajustabilidad a escala temporal se puede lograr dentro de una capa definiendo un grupo de imágenes con un nivel temporal particular como una subcapa.

[0048] Para cada capa respectiva del flujo de bits, los datos en una capa inferior pueden decodificarse sin referencia a los datos en cualquier capa superior. En la codificación de vídeo ajustable a escala, por ejemplo, los datos en una

capa de base pueden decodificarse sin referencia a los datos en una capa de mejora. En general, las unidades de NAL solo pueden encapsular datos de una única capa. Por tanto, las unidades de NAL que encapsulan los datos de la capa restante más alta del flujo de bits pueden eliminarse del flujo de bits sin afectar a la decodificabilidad de los datos en las capas restantes del flujo de bits. En la codificación de vista múltiple, las capas superiores pueden incluir componentes de vista adicionales. En la SHVC, las capas superiores pueden incluir datos de mejora de la razón entre señal y ruido (SNR), datos de mejora espacial y/o datos de mejora temporal. En la MV-HEVC y la SHVC, una capa puede denominarse "capa base" si un decodificador de vídeo puede decodificar imágenes en la capa sin hacer referencia a los datos de cualquier otra capa. La capa base puede ajustarse a la especificación de base de la HEVC (por ejemplo, Rec. UIT-T H.265 | ISO / IEC 23008-2).

[0049] En la codificación de vídeo escalable, las capas distintas a la capa base pueden denominarse "capas de mejora" y pueden proporcionar información que mejore la calidad visual de los datos de vídeo decodificados a partir del flujo de bits. La codificación de vídeo ajustable a escala puede mejorar la resolución espacial, la razón entre señal y ruido (es decir, la calidad) o la tasa temporal. En la codificación de vídeo ajustable a escala (por ejemplo, la SHVC), una "representación de capa" puede ser una representación codificada de una capa espacial en una única unidad de acceso. Para facilitar la explicación, esta divulgación puede referirse a componentes de vista y/o representaciones de capa como "componentes de vista / representaciones de capa" o simplemente "imágenes".

[0050] La codificación de vista múltiple da soporte a la predicción entre vistas. La predicción entre vistas es similar a la interpredicción utilizada en la HEVC y puede usar los mismos elementos sintácticos. Sin embargo, cuando un codificador de vídeo realiza la predicción entre vistas en una unidad de vídeo actual (tal como una PU), el codificador de vídeo 20 puede usar, como una imagen de referencia, una imagen que está en la misma unidad de acceso que la unidad de vídeo actual, pero en una vista diferente. Por lo contrario, la interpredicción convencional solo usa imágenes en diferentes unidades de acceso como imágenes de referencia.

[0051] En la codificación de vista múltiple, una vista puede denominarse "vista de base" si un decodificador de vídeo (por ejemplo, el decodificador de vídeo 30) puede decodificar imágenes en la vista sin referencia a imágenes en ninguna otra vista. Cuando se codifica una imagen en una de las vistas no de base, un codificador de vídeo (tal como el codificador de vídeo 20 o el decodificador de vídeo 30) puede agregar una imagen a una lista de imágenes de referencia si la imagen está en una vista diferente pero dentro de la misma instancia de tiempo (es decir, unidad de acceso) que la imagen que el codificador de vídeo está codificando actualmente. Al igual que otras imágenes de referencia de interpredicción, el codificador de vídeo puede insertar una imagen de referencia de predicción entre vistas en cualquier posición de una lista de imágenes de referencia.

[0052] Por ejemplo, las unidades de NAL pueden incluir encabezados (es decir, encabezados de unidades de NAL) y cargas útiles (por ejemplo, las RBSP). Los encabezados de la unidad de NAL pueden incluir elementos sintácticos `nuh_reservado_cero_6bits`, que también pueden denominarse elementos sintácticos `nuh_layer_id`. Las unidades de NAL que tienen elementos sintácticos `nuh_reservado_cero_6bit` que especifican valores diferentes pertenecen a diferentes "capas" de un flujo de bits. Por lo tanto, en la codificación de vista múltiple, la MV-HEVC, la SVC o la SHVC, el elemento sintáctico `nuh_layer_id` de la unidad de NAL especifica un identificador de capa (es decir, un ID de capa) de la unidad de NAL. El elemento sintáctico `nuh_layer_id` de una unidad de NAL es igual a 0 si la unidad de NAL se refiere a una capa de base en la codificación de vista múltiple, la MV-HEVC o la SHVC. Los datos en una capa de base de un flujo de bits pueden decodificarse sin referencia a los datos en cualquier otra capa del flujo de bits. Si la unidad de NAL no se refiere a una capa de base en la codificación de vista múltiple, la MV-HEVC o la SHVC, el elemento sintáctico `nuh_layer_id` puede tener un valor distinto de cero. En la codificación de vista múltiple, las diferentes capas de un flujo de bits pueden corresponder a diferentes vistas. En la SVC o la SHVC, las capas distintas a la capa de base pueden denominarse "capas de mejora" y pueden proporcionar información que mejora la calidad visual de los datos de vídeo decodificados a partir del flujo de bits.

[0053] Además, algunas imágenes dentro de una capa se pueden decodificar sin referencia a otras imágenes dentro de la misma capa. Por lo tanto, las unidades de NAL que encapsulan datos de ciertas imágenes de una capa pueden eliminarse del flujo de bits sin afectar la capacidad de decodificación de otras imágenes en la capa. La eliminación de unidades de NAL que encapsulan datos de tales imágenes puede reducir la velocidad de tramas del flujo de bits. Un subconjunto de imágenes dentro de una capa que puede decodificarse sin referencia a otras imágenes dentro de la capa se puede denominar en el presente documento una "subcapa" o una "subcapa temporal". Por lo tanto, un tipo de dimensión ajustable a escala es la dimensión temporal.

[0054] Por ejemplo, en la ajustabilidad a escala temporal, un conjunto de datos de vídeo puede dar soporte a varias velocidades de trama o velocidades de reproducción, por ejemplo, 15 tramas por segundo (FPS), 30 FPS, 60 FPS y 120 FPS. Un nivel temporal dado puede incluir todas las imágenes en ese nivel y en niveles inferiores. Por ejemplo, continuando con el ejemplo anterior, un nivel temporal de 0 puede corresponder a 15 FPS, un nivel temporal de 1 puede incluir imágenes del nivel temporal 0, así como imágenes en el nivel temporal 1 para dar soporte a 30 FPS, un nivel temporal de 2 puede incluir imágenes de los niveles temporales 0 y 1, así como imágenes en el nivel temporal 2 para dar soporte a 60 FPS, y así sucesivamente.

[0055] Un identificador temporal, o IDtemporal, puede ser señalado como representativo del nivel temporal al que pertenece una imagen en particular. Por ejemplo, las unidades de NAL pueden incluir elementos sintácticos temporal_id. El elemento sintáctico temporal_id de una unidad de NAL especifica un identificador temporal de la unidad de NAL. El identificador temporal de una unidad de NAL identifica una subcapa temporal con la que está asociada la unidad de NAL. Por lo tanto, cada subcapa temporal de un flujo de bits puede asociarse a un identificador temporal diferente. Si el identificador temporal de una primera unidad de NAL es menor que el identificador temporal de una segunda unidad de NAL, los datos encapsulados por la primera unidad de NAL pueden decodificarse sin referencia a los datos encapsulados por la segunda unidad de NAL.

[0056] Un flujo de bits puede estar asociado a una pluralidad de puntos de operación. En algunos ejemplos, cada punto de operación de un flujo de bits puede asociarse a un conjunto de identificadores de capa (es decir, un conjunto de valores nuh_reservado_cero_6bits o valores nuh_layer_id) y un identificador temporal. El conjunto de identificadores de capa se puede indicar como OpLayerIdSet y el identificador temporal se puede indicar como IDTemporal. Si el identificador de capa de una unidad de NAL se encuentra en el conjunto de identificadores de capa de un punto de operación y el identificador temporal de la unidad de NAL es menor o igual que el identificador temporal del punto de operación, la unidad de NAL está asociada al punto de operación. Por lo tanto, un punto de operación puede ser un flujo de bits creado a partir de otro flujo de bits mediante la operación del proceso de extracción del subflujo de bits con este otro flujo de bits, un IDTemporal máximo de destino y una lista de identificadores de capa de destino como entradas para el proceso de extracción del subflujo de bits. El punto de funcionamiento puede incluir cada unidad de NAL que esté asociada al punto de funcionamiento. En algunos ejemplos, el punto de operación no incluye unidades de NAL de VCL que no estén asociadas al punto de operación.

[0057] Como se ha indicado anteriormente, un punto de operación describe un subconjunto de capas (por ejemplo, vistas) de un conjunto completo de capas de datos de vídeo de múltiples capas. El punto de operación también puede identificar las capas de salida de destino, es decir, las capas para las cuales se han de emitir datos (por ejemplo, imágenes decodificadas). En algunos casos, los datos de una capa pueden incluirse en un punto de operación solo para su uso como capa de referencia (por ejemplo, solo para la predicción entre capas o entre vistas de la capa o vista que se mostrará, donde en la capa de referencia o la vista no se muestra) y, por lo tanto, tal capa no se consideraría una capa de salida de destino.

[0058] La HEVC y otras normas de codificación de vídeo especifican perfiles, gradas y niveles. Los perfiles, gradas y niveles especifican restricciones sobre los flujos de bits y, por lo tanto, límites sobre las capacidades necesarias para decodificar los flujos de bits. Los perfiles, gradas y niveles también se pueden utilizar para indicar puntos de interoperabilidad entre implementaciones de decodificadores individuales. Cada perfil especifica un subconjunto de características y herramientas algorítmicas presentes en una norma de codificación de vídeo. Por lo tanto, un "perfil" es un subconjunto de una sintaxis de flujo de bits completa que se especifica mediante una norma de codificación de vídeo aplicable. Los codificadores de vídeo no están obligados a utilizar todas las características con soporte en un perfil. Cada nivel de una grada puede especificar un conjunto de límites sobre los valores que pueden tener los elementos y las variables sintácticas. Así, un "nivel" corresponde a las limitaciones del consumo de recursos del decodificador, tales como, por ejemplo, memoria y cálculos del decodificador, que están relacionados con la resolución de las imágenes, la velocidad de bits y la velocidad de procesamiento de bloques. El mismo conjunto de definiciones de grada y nivel puede usarse con todos los perfiles, pero las implementaciones individuales pueden prestar soporte a una grada diferente y, dentro de una grada, un nivel diferente para cada perfil con soporte. Para cualquier perfil dado, un nivel de una grada generalmente puede corresponder a una carga de procesamiento y capacidad de memoria del decodificador en particular. Las capacidades de los decodificadores de vídeo se pueden especificar en términos de la capacidad de decodificar flujos de vídeo que se ajusten a las restricciones de perfiles, gradas y niveles particulares. Para cada perfil, la grada y el nivel con soporte para ese perfil también se pueden expresar. Es posible que algunos decodificadores de vídeo no puedan decodificar perfiles, gradas o niveles particulares.

[0059] Esta divulgación describe técnicas que pueden mejorar el descriptor del punto de operación de la HEVC en el flujo de transporte (TS) del MPEG-2 para el transporte de flujos de bits de extensión de la HEVC. Por ejemplo, de acuerdo a una técnica de esta divulgación, el multiplexador 21 y / o el demultiplexador 29 pueden configurarse para transportar datos de vídeo (es decir, enviar o recibir datos de vídeo) que están codificados de acuerdo a una norma de codificación de vídeo, tal como la HEVC, una extensión de una norma de codificación de vídeo (por ejemplo, extensiones de la norma HEVC, tales como la SHVC o la MV-HEVC) u otras normas de codificación de vídeo aún no desarrolladas. En general, el multiplexador 21 puede encapsular datos de vídeo codificados para formar un flujo de datos, por ejemplo, esencialmente de acuerdo con los sistemas del MPEG-2 y las técnicas de esta divulgación, mientras que el demultiplexador 29 puede recibir y desencapsular datos encapsulados, por ejemplo, datos de vídeo codificados de acuerdo con una extensión de una norma de codificación de vídeo.

[0060] La más reciente especificación de TS del MPEG-2 es la recomendación ITU-T H.222.0, versión de junio de 2012, en la que se proporciona el soporte de la AVC y de las extensiones de la AVC. También se ha desarrollado una enmienda de TS del MPEG-2 para la HEVC. El más reciente documento, "Text of ISO/IEC 13818-1: 2013/Final Draft Amendment 3 – Transport of HEVC video over MPEG-2 Systems,[" Texto de la ISO / IEC 13818-1: 2013 / Enmienda de Borrador Final 3 - Transporte de vídeo de la HEVC sobre sistemas del MPEG-2"], está disponible en el documento MPEG w13656, julio de 2013. Recientemente, se ha iniciado una enmienda del TS del MPEG-2 para el transporte de

la HEVC en capas, denominada TS del MPEG-2 para la L-HEVC. El más reciente documento es "Text of ISO/IEC 13818-1:2013 / DAM 3 – Carriage of Layered HEVC" ["Texto de la ISO / IEC 13818-1: 2013 / DAM 3 - Transporte de la HEVC en capas"], en adelante "DAM3". El término "HEVC en capas" se refiere a las extensiones de la norma HEVC que usan múltiples capas, tales como la SHVC, la MV-HEVC y la 3D-HEVC.

[0061] La especificación de los sistemas del MPEG-2 describe cómo los flujos de datos de multimedia (vídeo y audio) comprimidos pueden multiplexarse junto con otros datos para formar un solo flujo de datos adecuado para la transmisión o almacenamiento digital. La especificación de sistemas del MPEG-2 describe un flujo elemental, que es un componente único, codificado digitalmente (posiblemente comprimido según el MPEG) de un programa (también a veces escrito "programa"). Por ejemplo, la parte de vídeo o audio codificado del programa puede ser un flujo elemental. Un flujo elemental se convierte primero en un flujo elemental empaquetado (PES) antes de ser multiplexado en un flujo de programa o un flujo de transporte. Dentro del mismo programa, se utiliza un elemento sintáctico `id_flujo` para distinguir los paquetes del PES que pertenecen a un flujo elemental de los de otro. Cada paquete del PES puede contener una o más unidades de NAL o una unidad de NAL puede dividirse entre múltiples paquetes del PES.

[0062] En la especificación de sistemas del MPEG-2, los flujos de programas y los flujos de transporte son dos multiplexos alternativos que se orientan a diferentes aplicaciones. Los flujos de programas están predispuestos para el almacenamiento y la visualización de un solo programa desde un servicio de almacenamiento digital, y un flujo de programas está concebido para su uso en entornos sin errores, ya que puede ser susceptible a errores.

[0063] Un flujo de programa incluye los flujos elementales que le pertenecen y generalmente contiene paquetes con paquetes de longitud variable. En un flujo de programa, los paquetes del PES que se obtienen de los flujos elementales contribuyentes se organizan en "fardos". Un fardo comprende una cabecera de fardo, una cabecera de sistema optativa y cualquier número de paquetes del PES tomados de cualquiera de los flujos elementales contribuyentes, en cualquier orden. El encabezado del sistema contiene un resumen de las características del flujo de programa, tales como: su máxima velocidad de datos; el número de flujos elementales de audio y vídeo que contribuyen; e información adicional de temporización. El decodificador de vídeo 30 puede usar la información contenida en un encabezado del sistema para determinar si el decodificador de vídeo tridimensional es capaz de decodificar el flujo de programa o no.

[0064] Los flujos de transporte están concebidos para la entrega simultánea de una serie de programas por canales potencialmente propensos a errores. Un flujo de transporte es un multiplexo ideado para aplicaciones de múltiples programas, tales como la difusión, de manera que un solo flujo de transporte pueda asimilar muchos programas independientes. Un flujo de transporte incluye una sucesión de paquetes de transporte, y cada uno de los paquetes de transporte tiene una longitud de 188 octetos. El uso de paquetes cortos de longitud fija significa que el flujo de transporte no es tan susceptible a errores como el flujo de programa. Además, a cada paquete de transporte de 188 octetos de longitud se otorga fácilmente protección adicional contra errores procesándolo a través de un proceso estándar de protección contra errores, tal como la codificación de Reed-Solomon. La mejora de la resistencia a errores del flujo de transporte significa que tiene una mejor oportunidad de sobrevivir a los canales propensos a errores que se encuentran en un entorno de difusión, por ejemplo. Podría parecer que el flujo de transporte es claramente el mejor de los dos multiplexos, con su acrecentada resistencia a los errores y su capacidad para transportar muchos programas simultáneos. Sin embargo, el flujo de transporte es un multiplexo más sofisticado que el flujo de programa y, por consiguiente, más difícil de crear y demultiplexar.

[0065] El primer octeto de un paquete de transporte es un octeto de sincronización que, en algunos casos, es `0x47`. Un único flujo de transporte puede llevar muchos programas diferentes, comprendiendo cada uno muchos flujos elementales empaquetados. Un campo de identificador de paquete (PID) se usa para distinguir los paquetes de transporte que contienen los datos de un flujo elemental de aquellos que llevan los datos de otros flujos elementales. En algunos casos, el PID es de 13 bits. Puede ser responsabilidad del multiplexor 21 garantizar que cada flujo elemental reciba un valor único de PID.

[0066] Aunque está claro, basándose en un valor de PID, a qué flujo elemental pertenece un paquete de transporte, el decodificador de vídeo 30 puede necesitar saber qué flujos elementales pertenecen a qué programa. Por consiguiente, un flujo de transporte comprende información específica de programa (PSI) para especificar explícitamente las relaciones entre los programas y los flujos elementales componentes. En otras palabras, el flujo de transporte puede incluir paquetes de transporte que contienen PSI.

[0067] La PSI puede incluir una tabla de asociación de programas (PAT). La tabla de asociación de programas incluye una lista completa de todos los programas disponibles en un flujo de transporte. En algunos ejemplos, la PAT siempre tiene el valor 0 del PID. Cada programa se enumera junto con el valor del PID de los paquetes de transporte que contienen la tabla de correlaciones de programa del programa.

[0068] Además, la PSI puede incluir una o más tablas de correlaciones de programa (PMT). Cada programa transportado en un flujo de transporte tiene una PMT asociada. La PAT puede especificar el valor del PID de los paquetes de transporte que contienen la PMT para un programa. La PMT para un programa proporciona detalles sobre el programa y los flujos elementales que comprenden el programa. Por ejemplo, la PMT para un programa con número de programa 3 puede especificar que el programa contiene paquetes de transporte con valores de PID 33, 57 y 60.

En este ejemplo, los paquetes de transporte con valores de PID iguales a 33 pueden incluir un flujo elemental que contiene datos de vídeo codificados, los paquetes de transporte con valores de PID iguales a 57 pueden incluir datos de audio en inglés y los paquetes de transporte con valores de PID iguales a 60 pueden incluir datos de audio en chino. Una PMT puede incluir detalles sobre más de un programa.

5
[0069] La PMT básica para un programa puede incluir algunos de los muchos descriptores especificados dentro de la especificación de sistemas del MPEG-2. Dichos descriptores transmiten información adicional sobre un programa o sus flujos elementales componentes. Los descriptores pueden incluir parámetros de codificación de vídeo, parámetros de codificación de audio, identificación de idioma, información de vistas panorámicas y barridos, detalles de acceso condicional, información de derechos de autor, etc. Una emisora u otro usuario puede definir descriptores privados adicionales si es necesario.

15
[0070] Los descriptores son independientes de los datos de vídeo codificados. Por lo tanto, un dispositivo, tal como un Elemento de Red Consciente de los Medios (MANE) o un decodificador de vídeo, puede ser capaz de usar un descriptor para realizar varias funciones en flujos de transporte y flujos de programas sin decodificar o analizar de otro modo datos de vídeo codificados. Por ejemplo, si los datos de vídeo están codificados usando la HEVC, el dispositivo no necesita ser configurado para decodificar los datos de vídeo codificados por la HEVC para usar el descriptor para realizar funciones particulares en flujos de transporte o programas. Por ejemplo, el dispositivo puede ser capaz de usar los descriptores como parte de un proceso para determinar si se remiten elementos de programa particulares a un dispositivo de destino o se decodifican flujos elementales particulares.

25
[0071] Los descriptores para un programa que tiene flujos elementales de componentes relacionados con el vídeo pueden incluir uno o más descriptores jerárquicos. Un descriptor jerárquico está diseñado para señalar la jerarquía de los subflujos de bits de diferentes flujos elementales. El descriptor jerárquico proporciona información que identifica los elementos de programa que contienen componentes de flujos de vídeo, de audio y privados, codificados jerárquicamente. Los flujos privados pueden incluir metadatos, tales como un flujo de información específica de programa. En general, un elemento de programa es uno de los datos o flujos elementales incluidos en un programa (es decir, un flujo elemental componente del programa). En los flujos de transporte del MPEG-2, los elementos de programa normalmente se empaquetan. En los flujos de programa del MPEG-2, los elementos de programa no están empaquetados.

35
[0072] En algunos casos, cada respectiva subcapa temporal de cada respectiva capa de un programa puede corresponder a un componente de programa diferente (por ejemplo, flujo elemental) del programa. En otros casos, cada capa respectiva de un programa corresponde a un flujo elemental diferente. Además, en algunos casos, dos o más flujos elementales pueden corresponder a diferentes subcapas temporales de la misma capa de un programa, mientras que un solo flujo elemental puede corresponder a todas las subcapas temporales de otra capa del mismo programa. Como se ha indicado anteriormente, los descriptores pueden incluir descriptores jerárquicos. Cada descriptor jerárquico respectivo proporciona información con respecto a un correspondiente componente de programa (es decir, flujo elemental). Por ejemplo, un descriptor jerárquico correspondiente a un flujo elemental particular puede incluir un elemento sintáctico que especifique otro flujo elemental necesario para decodificar datos de vídeo del flujo elemental particular. En un ejemplo, un descriptor jerárquico correspondiente a un flujo elemental para una subcapa temporal particular puede incluir un elemento sintáctico que especifique una subcapa temporal integrada, necesaria para decodificar la subcapa temporal particular.

45
[0073] La subcapa temporal integrada se puede "integrar" en el sentido de que la decodificación de la subcapa temporal integrada es necesaria para la decodificación exitosa de la subcapa temporal particular. Además, el descriptor jerárquico puede incluir elementos sintácticos que especifican si una subcapa temporal correspondiente al descriptor jerárquico proporciona ajustabilidad temporal a escala (por ejemplo, aumenta la velocidad de tramas) con respecto a la subcapa temporal integrada, proporciona ajustabilidad espacial a escala (por ejemplo, aumenta la resolución de la imagen)) en relación con la subcapa temporal integrada, proporciona ajustabilidad de calidad a escala (por ejemplo, mejora la calidad o fidelidad entre señal y ruido) con respecto a la subcapa temporal integrada, y así sucesivamente.

55
[0074] La Tabla 2-49, a continuación, especifica la sintaxis de un descriptor jerárquico según se define en DAM3.

Tabla 2-49 - Descriptor jerárquico

Sintaxis	Nº de bits	Mnemónico
hierarchy_descriptor () {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
no_view_scalability_flag	1	bslbf
no_temporal_scalability_flag	1	bslbf
no_spatial_scalability_flag	1	bslbf
no_quality_scalability_flag	1	bslbf
hierarchy_type	4	uimsbf
reserved	2	bslbf
hierarchy_layer_index	6	uimsbf
tref_present_flag	1	bslbf
reserved	1	bslbf
hierarchy_embedded_layer_index	6	uimsbf
reserved	2	bslbf
hierarchy_channel	6	uimsbf
}		

[0075] En la Tabla 2-49, el **hierarchy_layer_index** es un campo de 6 bits que define un índice único del elemento de programa asociado en una tabla de jerarquías de capas de codificación. Los índices serán únicos dentro de una sola definición de programa. Para los sub-flujos de bits de vídeo de los flujos de vídeo de la HEVC que se ajustan a uno o más perfiles definidos en el Anexo F de la Rec. ITU-T H.265 | ISO / CEI 23008-2, este es el índice del elemento del programa, que se asigna de manera que el orden de los flujos de bits sea correcto si las capas de dependencia asociadas de los sub-flujos de bits de vídeo de la misma unidad de acceso de la HEVC se vuelven a ensamblar en orden creciente del **hierarchy_layer_index**. En otras palabras, el **hierarchy_layer_index** de un descriptor jerárquico identifica el elemento de programa (es decir, el flujo elemental) que corresponde al descriptor jerárquico.

[0076] Además, en la Tabla 2-49, el **hierarchy_embedded_layer_index** es un campo de 6 bits que define el **hierarchy_layer_index** del elemento de programa al que se debe acceder y que debe estar presente en orden de decodificación antes de decodificar el flujo elemental asociado a este **hierarchy_descriptor**. El **hierarchy_embedded_layer_index** está indefinido si el valor de **hierarchy_type** es 15. En otras palabras, el **hierarchy_embedded_layer_index** de un descriptor jerárquico identifica un flujo elemental del cual depende el flujo elemental correspondiente al descriptor jerárquico.

[0077] Además de los uno o más descriptores jerárquicos, los descriptores señalados en un flujo de transporte o programa del MPEG-2 pueden incluir uno o más descriptores de extensión jerárquica. Cada respectivo descriptor de extensión jerárquica en el flujo de o transporte o de programa del MPEG-2 puede corresponder a un flujo elemental respectivo. Cada descriptor de extensión jerárquica puede proporcionar información adicional con respecto a un flujo elemental correspondiente. Como se ha indicado anteriormente, un flujo elemental puede corresponder a una o más subcapas temporales o puede corresponder a una capa como una totalidad. En otras palabras, un flujo elemental puede incluir datos de vídeo codificados de una o más subcapas temporales o puede corresponder a una capa como una totalidad. Por lo tanto, un descriptor de extensión jerárquica puede corresponder a un flujo elemental correspondiente a una o más subcapas temporales o puede corresponder a una capa como una totalidad.

[0078] Cuando un descriptor de extensión jerárquica está presente, el descriptor de extensión jerárquica se usa para especificar las dependencias del flujo elemental correspondientes al flujo elemental. Por ejemplo, un descriptor de extensión jerárquica correspondiente a un flujo elemental particular puede indicar múltiples flujos elementales que deben decodificarse para decodificar con éxito el flujo elemental correspondiente al descriptor de extensión jerárquica. Por lo contrario, un descriptor jerárquico solo es capaz de indicar un flujo elemental requerido para la decodificación exitosa de un flujo elemental correspondiente al descriptor jerárquico. La tabla 2-103decimens, a continuación, indica una sintaxis de un descriptor de extensión jerárquica, como se especifica en DAM3.

Tabla 2-103decimens - Descriptor de la extensión de la jerarquía de la HEVC

Sintaxis	Nº de bits	Mnemónico
HEVC_hierarchy_extension_descriptor() {		
extension_dimension_bits	16	bslbf
hierarchy_layer_index	6	uimsbf
temporal_id	3	uimsbf
nuh_layer_id	6	uimsbf
tref_present_flag	1	bslbf
reserved	2	bslbf
num_embedded_layers	6	uimsbf
reserved	2	bslbf
hierarchy_channel	6	uimsbf
for (i = 0 ; i < num_embedded_layers ; i++) {		
reserved	2	bslbf
hierarchy_ext_embedded_layer_index[i]	6	uimsbf
}		
}		

5 [0079] En la Tabla 2-103decimens, el **hierarchy_layer_index** es un campo de 6 bits que define un índice único del elemento de programa asociado en una tabla de jerarquías de capas de codificación. Los índices serán únicos dentro de una sola definición de programa. Para los sub-flujos de bits de vídeo de los flujos de vídeo de la HEVC que se ajustan a uno o más perfiles definidos en el Anexo G o H de la Rec. H.265 | ISO / IEC 23008-2, éste es el índice de elemento de programa, que se asigna de tal manera que el orden del flujo de bits será correcto si las capas de dependencia asociadas de los sub-flujos de bits de vídeo de la misma unidad de acceso se reensamblan en orden creciente del **hierarchy_layer_index**. Por lo tanto, el **hierarchy_layer_index** de un descriptor de extensión jerárquica identifica el flujo elemental correspondiente al descriptor de extensión jerárquica.

10 [0080] El **nuh_layer_id** es un campo de 6 bits que especifica el **nuh_layer_id** más alto de las unidades de NAL en el flujo elemental asociado a este HEVC_hierarchy_extension_descriptor().

15 [0081] El **temporal_id** es un campo de 3 bits que especifica el **ldtemporal** más alto de las unidades de NAL en el flujo elemental asociado a este HEVC_hierarchy_extension_descriptor().

20 [0082] El **num_embedded_layers** es un campo de 6 bits que especifica el número de elementos de programa dependientes directos a los que se debe acceder y que deben estar presentes en el orden de decodificación antes de decodificar el flujo elemental asociado a este HEVC_hierarchy_extension_descriptor().

25 [0083] El **hierarchy_ext_embedded_layer_index** es un campo de 6 bits que define el **hierarchy_layer_index** del elemento de programa al que se necesita acceder y que debe estar presente en el orden de decodificación antes de la decodificación del flujo elemental asociado a este **hierarchy_extension_descriptor**. Este campo está indefinido si el valor de **hierarchy_type** es 15.

30 [0084] Además del descriptor de jerarquía y de los descriptores de extensión de jerarquía, un flujo de transporte puede incluir un descriptor de punto de operación de la HEVC. Los puntos de operación de la HEVC se señalizan en un descriptor de punto de operación de la HEVC especificando las referencias a los flujos elementales correspondientes a las capas en los puntos de operación. Como se describe en DAM3, el descriptor del punto de operación de la HEVC proporciona un procedimiento para indicar un perfil y un nivel para uno o más puntos de operación de la HEVC. El dispositivo de destino 14 puede usar los descriptores de punto de operación incluidos en un flujo de bits para seleccionar uno de los puntos de operación a decodificar y finalmente presentar (por ejemplo, exhibir) a un usuario. En lugar de pasar datos para todas las vistas o capas al decodificador de vídeo 30 al recibirlos, el dispositivo de destino 14 puede enviar solo las vistas de un punto de operación seleccionado al decodificador de vídeo 30. Por ejemplo, el dispositivo de destino 14 puede descartar datos para vistas que no se decodificarán. Adicional o alternativamente, un dispositivo de red intermedio (por ejemplo, un elemento de red consciente de los medios (MANE)) puede descartar datos para vistas o capas que no corresponden a un punto de operación solicitado (por ejemplo, para utilizar mejor el ancho de banda). El dispositivo de destino 14 puede seleccionar un punto de operación basándose en la calidad más alta con soporte en uno de los puntos de operación para un flujo de bits y / o basándose en una cantidad disponible de ancho de banda de red.

[0085] La tabla 2-103noniens a continuación indica la sintaxis del descriptor del punto de operación de la HEVC, como se define en DAM3.

Tabla 2-103noniens - Descriptor del punto de operación de la HEVC

Sintaxis	Nº de bits	Mnemónico
HEVC_operation_point_descriptor() {		
num_ptl	8	uimsbf
for (i = 0; i < num_ptl; i++, i++) {		
profile_tier_level_info[i]	96	bslbf
}		
operation_points_count	8	uimsbf
for (i = 0; i < operation_points_count; i++) {		
target_ols[i]	8	uimsbf
ES_count[i]	8	uimsbf
<EMPHASIS>for (j = 0; j < ES_count;		
j++) {		
output_layer_flag[i][j]	1	bslbf
prepend_dependencies[i][j]	1	bslbf
ES_reference[i][j]	6	uimsbf
ptl_ref_idx[i][j]	8	uimsbf
}</EMPHASIS>		
reserved	1	bslbf
avg_bit_rate_info_flag[i]	1	bslbf
max_bit_rate_info_flag[i]	1	bslbf
constant_frame_rate_info_idc[i]	2	uimsbf
applicable_temporal_id[i]	3	uimsbf
if (constant_frame_rate_info_idc[i] > 0)		
{		
reserved	4	bslbf
frame_rate_indicatory[i]	12	uimsbf
}		
if (avg_bit_rate_info_flag[i] == '1') {		
avg_bit_rate[i]	24	uimsbf
}		
if (max_bit_rate_info_flag[i] == '1') {		
max_bit_rate[i]	24	uimsbf
}		
}		
}		

5 [0086] de punto de operación de la HEVC). Los siguientes párrafos describen dos de los problemas en el diseño actual del descriptor de punto de operación de la HEVC definido en DAM3.

10 [0087] En primer lugar, como se muestra en la Tabla 2-103noniens, el descriptor del punto de operación de la HEVC incluye elementos sintácticos *prepend_dependencies[i][j]*. Sin embargo, en DAM3, falta la semántica de *prepend_dependencies[i][j]* igual a 0. Además, en el diseño de DAM3, es posible que un flujo elemental (por ejemplo, un flujo elemental correspondiente a una capa) se incluya dos veces en la lista de flujos elementales para el punto de operación de destino. El problema de más de una inclusión puede ocurrir en cualquiera de los siguientes escenarios:

15 1. Cuando una capa se señala explícitamente como parte de un punto de operación al señalar su valor de *ES_reference[i][j]* y al mismo tiempo también es una capa de referencia de otra capa que tiene el valor de *prepend_dependencies[i][j]* igual a 1.

20 2. Cuando una capa es una capa de referencia para otras dos capas en el mismo punto de operación que tienen, ambas, *prepend_dependencies[i][j]* igual a 1.

25 [0088] Un segundo problema en el diseño en DAM3 para el descriptor del punto de operación de la HEVC puede ser que, para un punto de operación, no todas las capas cuya decodificación sea requerida por el decodificador de vídeo 30 para el punto de operación de la HEVC deben señalarse explícitamente, porque algunas capas (los flujos elementales) pueden depender de otras y dicha información está presente en otro lugar (por ejemplo, en el descriptor jerárquico o el descriptor de extensión jerárquica). Cuando el valor del elemento sintáctico *prepend_dependencies[i][j]* es igual a 1, se pueden obtener otras capas de las que depende la j-ésima capa y, por lo tanto, no se señalan explícitamente para el i-ésimo punto de operación de la HEVC. Sin embargo, en el punto de operación de la HEVC, una capa obtenida no está presente en el bucle de las capas explícitas (como lo indica el texto entre las etiquetas "<EMPHASIS>" y "</EMPHASIS>" en la Tabla 2-103noniens en lo que antecede). Por lo tanto, actualmente no hay ninguna forma de conocer los valores de los siguientes elementos sintácticos para una capa obtenida:

output_layer_flag[i][j] y *ptl_ref_idx[i][j]*, aunque el flujo elemental de la capa obtenida puede identificarse mediante el descriptor jerárquico y / o el descriptor de extensión jerárquica. En otras palabras, cuando las capas de señalización de un punto de operación con referencias a flujos elementales y varias capas no se señalizan explícitamente, sino que se obtienen (cuando el valor del elemento sintáctico *prepend_dependencies[i][j]* es 1), la información, tal como si una capa es una capa de salida y el perfil, la grada y el nivel para esas capas obtenidas, falta para estas capas obtenidas.

[0089] Un punto de operación de salida en la HEVC en capas se define en función de un conjunto de capas de salida. Sin embargo, saber si una capa es una capa de salida puede ser requerido para la definición de conformidad en la HEVC en capas y cada capa necesaria (es decir, una capa que es bien una capa de salida o bien una capa que es mencionada directa o indirectamente por una capa de salida dentro de un punto de operación, o ambas) se asociará a un conjunto de información de perfil, grada y nivel (PTL) de acuerdo a la especificación de la MV-HEVC / SHVC. Por lo tanto, puede ser necesario saber si una capa es o no una capa de salida de destino. Por ejemplo, puede ser necesario conocer el valor del elemento sintáctico *output_layer_flag[i][j]*. Sin embargo, el elemento sintáctico *output_layer_flag[i][j]* también puede no estar presente en el flujo de bits para las capas obtenidas, porque el bucle 'para', incluido en las etiquetas "<EMPHASIS>" y "</EMPHASIS>" en la Tabla 2-103noniens anterior, solo puede ser aplicable para capas explícitamente señalizadas.

[0090] Además, la HEVC en capas, por diseño, requiere que esté presente la información de perfil, grada y nivel para cada capa necesaria (es decir, una capa que sea una capa de salida de destino, o necesaria para decodificar una capa de salida de destino). El diseño de DAM3 carece de tal información para las capas obtenidas. Además, para una capa innecesaria (es decir, una capa que no es una capa necesaria), la información de PTL no se señala según la especificación de codificación y, por lo tanto, en el diseño de transporte del Flujo de Transporte, señalar siempre PTL para capas innecesarias sería problemático, ya que hay no hay ninguna información de PTL para estas capas por señalar.

[0091] A continuación se describen técnicas ejemplares que pueden abordar las cuestiones anteriores. Sin embargo, no debería interpretarse como un requisito que las técnicas que se describen a continuación aborden las cuestiones anteriores. Las técnicas ejemplares pueden ser aplicadas por el codificador de vídeo 20, el multiplexador 21, el demultiplexador 29, un dispositivo intermedio (por ejemplo, un MANE) u otro dispositivo como parte de la producción del flujo de bits que el decodificador de vídeo 30 recibe eventualmente, o pueden ser aplicadas por el decodificador de vídeo 30 u otro dispositivo como parte de la decodificación de datos de vídeo para producir las imágenes a exhibir. Algunas de las técnicas se pueden aplicar de forma independiente y algunas de ellas se pueden aplicar en combinación. Además, aunque la divulgación describe los descriptores de puntos de operación de la HEVC y, de lo contrario, se refiere a la HEVC, las técnicas de esta divulgación pueden ser aplicables a normas de codificación de vídeo que no sean la HEVC.

[0092] De acuerdo con una primera técnica de esta divulgación, se señala un indicador para cada flujo elemental (ES) / capa en el punto de operación de la HEVC para indicar si el ES / la capa es una capa necesaria o no. En esta divulgación, las referencias a ES / capa o capa / ES se refieren a un flujo elemental correspondiente a una capa o a una subcapa temporal de la capa. Por lo tanto, los datos que identifican un flujo elemental se pueden usar para identificar la capa correspondiente y viceversa. Por ejemplo, el codificador de vídeo 20, el multiplexador 21 o un dispositivo intermedio pueden emitir (por ejemplo, señalar) un indicador (por ejemplo, *necessary_layer_flag[i][k]*, descrito a continuación) en el flujo de bits para cada flujo elemental o capa en el punto de operación, indicando si el flujo elemental, o la capa, es necesario o no. El decodificador de vídeo 30 puede recibir un indicador de este tipo y utilizar el indicador para decodificar el flujo de bits. En otras palabras, el decodificador de vídeo 30 puede decodificar el punto de operación basándose en el indicador recibido.

[0093] De acuerdo con una segunda técnica de esta divulgación, la semántica de *prepend_dependencies[i][j]* se actualiza para aclarar que la inclusión de un ES / una capa en la lista de flujos elementales para un punto de operación se aplicará solo cuando el ES no esté aún presente en la lista. Por ejemplo, se agrega la semántica de *prepend_dependencies[i][j]* igual a 0 y se aclara que se incluirá una capa / un flujo elemental en la lista de flujos elementales para el punto de operación de destino solo si la capa / el flujo elemental no está presente aún en la lista.

[0094] Así, en algunos ejemplos, un dispositivo, tal como el codificador de vídeo 20, genera un primer descriptor, tal como un descriptor de punto de operación. Además, el dispositivo puede generar un segundo descriptor, tal como un descriptor jerárquico o un descriptor de extensión jerárquica. Como parte de la generación del primer descriptor, el dispositivo puede incluir un primer elemento sintáctico y un segundo elemento sintáctico en el primer descriptor. En al menos algunos de tales ejemplos, una lista de flujos elementales es una lista de flujos elementales que forman parte del punto de operación y el segundo descriptor tiene un valor de índice de capa jerárquica igual a un valor del segundo elemento sintáctico. Un primer valor del primer elemento sintáctico especifica que un flujo elemental indicado por el segundo elemento sintáctico, cuando no esté presente en la lista de flujos elementales, se agregará a la lista de flujos elementales, y un flujo elemental indicado por un índice de capa integrada en el segundo descriptor, cuando no esté presente en la lista de flujos elementales, se agregará a la lista de flujos elementales. Un segundo valor del primer elemento sintáctico especifica que el flujo elemental indicado por el segundo elemento sintáctico, cuando no esté presente en la lista de flujos elementales, se agregará a la lista de flujos elementales, pero no el flujo elemental indicado

por el índice de capa integrada en el segundo descriptor. El dispositivo puede incluir el primer descriptor y el segundo descriptor en un flujo de transporte.

5 **[0095]** En un ejemplo similar, un dispositivo, tal como el decodificador de vídeo 30, recibe un flujo de transporte que incluye un primer descriptor, un segundo descriptor y una pluralidad de flujos elementales. El primer descriptor puede ser un descriptor para un punto de operación de los datos de vídeo. El segundo descriptor puede ser uno entre: un descriptor jerárquico y un descriptor de extensión jerárquica. El dispositivo puede decodificar un primer elemento sintáctico y un segundo elemento sintáctico en el primer descriptor. En al menos algunos de tales ejemplos, una lista de flujos elementales es una lista de flujos elementales del flujo de transporte que forman parte del punto de operación y el segundo descriptor tiene un valor de índice de capa jerárquica igual a un valor del segundo elemento sintáctico. Además, en dichos ejemplos, un primer valor del primer elemento sintáctico especifica que: se agregará un flujo elemental, indicado por el segundo elemento sintáctico, cuando no esté presente en la lista de flujos elementales, en la lista de flujos elementales, siendo el flujo elemental indicado por el segundo elemento sintáctico uno entre la pluralidad de flujos elementales, y se agregará un flujo elemental, indicado por un índice de capa integrada en el segundo descriptor, cuando no esté presente en la lista de flujos elementales, a la lista de flujos elementales, siendo el flujo elemental indicado por el índice de capa integrada en el segundo descriptor uno entre la pluralidad de flujos elementales. En respuesta a la determinación de que el primer elemento sintáctico tiene un segundo valor diferente al primer valor, el dispositivo puede agregar el flujo elemental indicado por el segundo elemento sintáctico, cuando no está presente en la lista de flujos elementales, en la lista de flujos elementales, pero no agregar el flujo elemental, indicado por el índice de capa integrada en el segundo descriptor, en la lista de flujos elementales.

25 **[0096]** De acuerdo con una tercera técnica de esta divulgación, se agrega una restricción al valor de *ES_reference[i][j]* de manera que el mismo flujo elemental no se señale explícitamente más de una vez para cualquier punto de operación en particular. En otras palabras, se sugiere además imponer una restricción para que el mismo flujo elemental no se señale explícitamente más de una vez para cualquier punto de operación en particular. Por ejemplo, el codificador de vídeo 20 o el dispositivo intermedio pueden configurarse de manera que el mismo flujo elemental no se señale explícitamente más de una vez.

30 **[0097]** Además, esta divulgación describe varias técnicas ejemplares que resuelven la segunda cuestión descrita anteriormente (es decir, el problema de no poder determinar los valores de ciertos elementos sintácticos para una capa obtenida, tal como el problema de la falta de indicador de capa de salida y el problema de información de PTL). En algunas de tales técnicas para resolver la segunda cuestión, se propone mantener la funcionalidad de obtener capas para un punto de operación, señalar la información faltante para las capas obtenidas, pero no señalar PTL para capas innecesarias.

35 **[0098]** En una primera técnica ejemplar para resolver la segunda cuestión, los elementos sintácticos de *prepend_dependencies[i][j]* se eliminan y todas las capas se señalan siempre de forma explícita.

40 **[0099]** En una segunda técnica ejemplar para resolver la segunda cuestión, se presentan dos bucles para las capas de un punto de operación en el descriptor de punto de operación de la HEVC. Un bucle contiene los elementos sintácticos *ES_reference[i][j]* y los elementos sintácticos *prepend_dependencies[i][j]* de las capas que están asociadas explícitamente a la referencia a un flujo elemental (*ES_reference[i][j]*). Otro bucle contiene los elementos sintácticos *output_layer_flag[i][k]* y los elementos sintácticos *ptl_ref_idx[i][k]* de todas las capas. El codificador de vídeo 20, el multiplexador 21, el demultiplexador 29, el decodificador de vídeo 30, un dispositivo intermedio y / u otro dispositivo pueden implementar estos bucles ejemplares para codificar o decodificar el descriptor del punto de operación de la HEVC.

50 **[0100]** En una instancia de la segunda técnica ejemplar para resolver la segunda cuestión, un dispositivo puede configurarse para realizar un primer bucle para generar o decodificar múltiples instancias (por ejemplo, conjuntos) de un primer elemento sintáctico (por ejemplo, *ES_reference[i][j]*) en un descriptor de punto de operación de la HEVC. El primer elemento sintáctico indica el valor de índice de capa jerárquica presente en el descriptor jerárquico o el descriptor de extensión jerárquica de la HEVC que identifica un flujo elemental de un punto de operación. En el primer bucle, el dispositivo también genera o decodifica múltiples instancias de un segundo elemento sintáctico (por ejemplo, *prepend_dependencies[i][j]*). El segundo elemento sintáctico indica a partir de qué estructuras deben incluirse los flujos elementales en una lista de flujos elementales para el punto de operación, como parte del rendimiento del primer bucle. Además, en este ejemplo, el dispositivo puede realizar un segundo bucle para generar o decodificar múltiples instancias de un tercer elemento sintáctico (por ejemplo, *output_layer_flag[i][k]*) que indica qué flujo elemental del punto de operación es una capa de salida. Además, en el segundo bucle, el dispositivo puede generar o decodificar múltiples instancias de un cuarto elemento sintáctico (por ejemplo, *ptl_ref_idx[i][k]*) que indica un índice para determinar un perfil, grada o nivel de un flujo elemental del punto de operación. En algunos casos, el dispositivo puede generar un flujo de transporte que incluya el descriptor del punto de operación de la HEVC o decodificar el punto de operación basándose en uno o más entre los elementos sintácticos primero, segundo, tercero y cuarto.

65 **[0101]** En una tercera técnica ejemplar para resolver la segunda cuestión, los dos bucles en la anterior segunda técnica ejemplar para resolver la segunda cuestión pueden fusionarse entre sí, con un indicador para cada capa que indica si la capa está señalizada explícita o implícitamente. Para este ejemplo, se supone que las capas, según lo

obtenido por el descriptor jerárquico o el descriptor de extensión jerárquica identificado por *ES_reference[i][j]*, formulan un cierto orden, de modo que es posible que las capas, según están ordenadas en el segundo bucle, correlacionen cada una de las capas señalizadas explícitamente con las capas según son señalizadas u obtenidas por la información en el primer bucle.

5 **[0102]** Una cuarta técnica ejemplar para resolver la segunda cuestión todavía habilita la funcionalidad mediante el elemento sintáctico *prepend_dependencias[i][j]*, pero, cuando se aplica, se deduce si una capa obtenida es una capa de salida con un valor predeterminado. Por ejemplo, cuando *prepend_dependencias[i][j]* es igual a 1, un dispositivo puede deducir que el valor de un elemento sintáctico *output_layer_flag[i][j]* es igual a un valor predeterminado. Dicho valor predeterminado, por ejemplo, 0 o 1, puede señalizarse para el punto de operación actual. El perfil, la grada y el nivel de una capa obtenida se configuran para que sean los mismos de la misma capa que ha sido explícitamente señalizada anteriormente en un punto de operación diferente.

15 **[0103]** Con el fin de no señalizar PTL para capas innecesarias, esta divulgación propone señalizar un indicador para cada ES / capa en el punto de operación de la HEVC, para indicar si el ES / la capa es una capa necesaria o no. La señalización utiliza un bit que, de lo contrario, se reservaría, por lo que no se introduce ninguna sobrecarga adicional y la señalización evita un proceso de obtención complicado que, de lo contrario, puede ser necesario especificar. Por ejemplo, un dispositivo (por ejemplo, el codificador de vídeo 20, el multiplexador 21, el demultiplexador 29, el decodificador de vídeo 30, un dispositivo intermedio u otro dispositivo) puede configurarse para determinar flujos o capas elementales de un punto de operación, determinar un indicador (por ejemplo, *necessary_layer_flag[i][k]*, descrito a continuación) para cada flujo o capa elemental en el punto de operación, en donde el indicador indica si un flujo, o capa, elemental respectivo es un flujo, o capa, necesario, y produce un flujo de bits que incluye los flujos elementales y los respectivos indicadores.

25 **[0104]** La figura 2 es un diagrama de bloques que ilustra un ejemplo del codificador de vídeo 20 que puede implementar técnicas para transportar datos de vídeo codificados de acuerdo con las extensiones de una norma de codificación de vídeo. Los datos de vídeo pueden incluir múltiples (por ejemplo, dos o más) capas de mejora para una capa de base, donde las capas de mejora pueden corresponder a diferentes dimensiones de ajustabilidad a escala.

30 **[0105]** En el ejemplo de la figura 2, el codificador de vídeo 20 incluye la memoria de datos de vídeo 49, la unidad de selección de modalidad 40, el almacén temporal de imágenes decodificadas (DPB) 64, el sumador 50, la unidad de procesamiento de transformación 52, la unidad de cuantización 54 y la unidad de codificación por entropía 56. A su vez, la unidad de selección de modalidad 40 incluye la unidad de compensación de movimiento 44, la unidad de estimación de movimiento 42, la unidad de intra-predicción 46 y la unidad de división 48. Para la reconstrucción de bloques de vídeo, el codificador de vídeo 20 incluye también la unidad de cuantización inversa 58, la unidad de transformación inversa 60 y el sumador 62.

35 **[0106]** La memoria de datos de vídeo 49 puede almacenar datos de vídeo a codificar por los componentes del codificador de vídeo 20. Los datos de vídeo almacenados en la memoria de datos de vídeo 49 se pueden obtener, por ejemplo, a partir del origen de vídeo 18. El DPB 64 puede ser una memoria de imágenes de referencia que almacena datos de vídeo de referencia para su uso en la codificación de datos de vídeo por el codificador de vídeo 20, por ejemplo, en modalidades de intracodificación o intercodificación. La memoria de datos de vídeo 49 y el DPB 64 pueden estar formados por cualquiera entre varios dispositivos de memoria, tales como memoria dinámica de acceso aleatorio (DRAM), incluyendo DRAM síncrona (SDRAM), RAM magneto-resistiva (MRAM), RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. La memoria de datos de vídeo 49 y el DPB 64 pueden ser proporcionados por el mismo dispositivo de memoria o por dispositivos de memoria independientes. En diversos ejemplos, la memoria de datos de vídeo 49 puede estar en un chip con otros componentes del codificador de vídeo 20, o fuera de chip con respecto a esos componentes.

40 **[0107]** Durante el proceso de codificación, el codificador 20 de vídeo recibe una trama o un fragmento de vídeo a codificar. La trama o el fragmento pueden dividirse 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 en una o más tramas de referencia, para proporcionar la predicción temporal. La unidad de intra-predicción 46, de forma alternativa, puede llevar a cabo la codificación intrapredictiva del bloque de vídeo recibido, con respecto a uno o más bloques contiguos en la misma trama o fragmento que el bloque a codificar, para proporcionar predicción espacial. El codificador de vídeo 20 puede llevar a cabo múltiples pases de codificación, por ejemplo, para seleccionar una modalidad de codificación adecuada para cada bloque de datos de vídeo.

45 **[0108]** Además, la unidad de división 48 puede dividir bloques de datos de vídeo en subbloques, basándose en la evaluación de los anteriores esquemas de división en los anteriores pases de codificación. Por ejemplo, la unidad de partición 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 velocidad-distorsión (por ejemplo, una optimización de velocidad-distorsión). La unidad de selección de modalidad 40 puede producir además una estructura de datos de árbol cuádruple, indicativa de la división de una LCU en las sub-CU. Las CU de nodos de hojas del árbol cuádruple pueden incluir una o más PU y una o más TU.

5 **[0109]** La unidad de selección de modalidad 40 puede seleccionar una de las modalidades de codificación, intra o inter, por ejemplo, basándose en los resultados de errores, y puede proporcionar el bloque intracodificado o intercodificado resultante al sumador 50 para generar datos de bloques residuales y al sumador 62 para reconstruir el bloque codificado para su uso en 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.

10 **[0110]** La estimación del movimiento, realizada por la unidad de estimación del movimiento 42, es el proceso de generación de vectores de movimiento, que estiman el movimiento de los bloques de vídeo. La unidad de estimación de movimiento 42 calcula un vector de movimiento para una PU de un bloque de vídeo en un fragmento intercodificado, comparando la posición de la PU con la posición de un bloque predictivo de una imagen de referencia. La imagen de referencia se puede seleccionar a partir de una primera lista de imágenes de referencia (ListalMgRef0) o una segunda lista de imágenes de referencia (ListalMgRef1), cada una de las cuales identifica una o más imágenes de referencia almacenadas en el DPB 64. La unidad de estimación del 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. 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 se está codificando, generando valores de diferencias de píxel, como se expone posteriormente. La unidad de selección de modalidad 40 también puede generar elementos sintácticos asociados a los bloques de vídeo y el fragmento de vídeo para su uso por el decodificador de vídeo 30 al decodificar los bloques de vídeo del fragmento de vídeo.

25 **[0111]** La unidad de intra-predicción 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. Después de seleccionar una modalidad de intra-predicción para un bloque, la unidad de intra-predicción 46 puede proporcionar información, indicativa de la modalidad de intra-predicción 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 intra-predicción seleccionada.

35 **[0112]** El codificador de vídeo 20 puede formar un bloque de vídeo residual restando los datos de predicción de la unidad de selección de modalidad 40 al bloque de vídeo original que se está codificando. El sumador 50 representa el componente o los componentes que realizan esta operación de resta. La unidad de procesamiento de transformación 52 puede aplicar una transformación, tal como una transformación discreta de 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 cuantización 54 puede cuantizar los coeficientes de transformación para reducir adicionalmente la tasa de bits. Después de la cuantización, la unidad de codificación por entropía 56 puede codificar por entropía los elementos sintácticos que indican los coeficientes de transformación cuantizados. Por ejemplo, la unidad de codificación por entropía 56 puede llevar a cabo la codificación de longitud variable adaptable al contexto (CAVLC), la codificación aritmética binaria adaptable al contexto (CABAC), la codificación aritmética binaria adaptable 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. 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 decodificador de vídeo 30) o archivarse para su posterior transmisión o recuperación.

50 **[0113]** 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 del píxel, por ejemplo, para su uso posterior como 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 del DPB 64. El sumador 62 puede agregar el bloque residual reconstruido al bloque de predicción de movimiento compensado por movimiento, producido por la unidad de compensación de movimiento 44 para producir un bloque de vídeo reconstruido para su almacenamiento en el DPB 64. El bloque de vídeo reconstruido puede ser utilizado por la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44, como bloque de referencia para intercodificar un bloque en una trama de vídeo subsiguiente.

60 **[0114]** En algunos ejemplos, el codificador de vídeo 20 de la figura 2 genera un primer descriptor, tal como un descriptor de punto de operación. Además, el codificador de vídeo 20 puede generar un segundo descriptor, tal como un descriptor jerárquico o un descriptor de extensión jerárquica. Como parte de la generación del primer descriptor, el codificador de vídeo 20 puede incluir un primer elemento sintáctico y un segundo elemento sintáctico en el primer descriptor. En al menos algunos de tales ejemplos, una lista de flujos elementales es una lista de flujos elementales que forman parte del punto de operación y el segundo descriptor tiene un valor de índice de capa jerárquica igual a un valor del segundo elemento sintáctico. Un primer valor del primer elemento sintáctico especifica que un flujo elemental indicado por el segundo elemento sintáctico, cuando no esté presente en la lista de flujos elementales, se agregará a la lista de flujos elementales, y un flujo elemental indicado por un índice de capa integrada en el segundo descriptor, cuando no esté presente en la lista de flujos elementales, se agregará a la lista de flujos elementales. Un segundo

valor del primer elemento sintáctico especifica que el flujo elemental indicado por el segundo elemento sintáctico, cuando no esté presente en la lista de flujos elementales, se agregará a la lista de flujos elementales, pero no el flujo elemental indicado por el índice de capa integrada en el segundo descriptor. El codificador de vídeo 20 puede incluir el primer descriptor y el segundo descriptor en un flujo de transporte.

[0115] La figura 3 es un diagrama de bloques que ilustra un ejemplo de decodificador de vídeo 30 que puede implementar técnicas para transportar datos de vídeo codificados de acuerdo con las extensiones de una norma de codificación de vídeo. En el ejemplo de la figura 3, el decodificador de vídeo 30 incluye una memoria de datos de vídeo 69, una unidad de decodificación por entropía 70, una unidad de compensación de movimiento 72, una unidad de intra-predicción 74, una unidad de cuantización inversa 76, una unidad de transformación inversa 78, un almacén temporal de imágenes decodificadas (DPB) 82 y un sumador 80.

[0116] La memoria de datos de vídeo 69 puede almacenar datos de vídeo, tales como un flujo de bits de vídeo codificado, para ser decodificado por los componentes del decodificador de vídeo 30. Los datos de vídeo almacenados en la memoria de datos de vídeo 69 pueden obtenerse, por ejemplo, a partir del medio legible por ordenador 16, por ejemplo, desde un origen de vídeo local, tal como una cámara, mediante comunicación de datos de vídeo por red cableada o inalámbrica, o accediendo a medios de almacenamiento físico de datos. La memoria de datos de vídeo 69 puede formar un almacén temporal de imágenes codificadas (CPB) que almacene datos de vídeo codificados a partir de un flujo de bits de vídeo codificado. El DPB 82 puede ser una memoria de imagen de referencia que almacena datos de vídeo de referencia para su uso en la decodificación de datos de vídeo por el decodificador de vídeo 30, por ejemplo, en modalidades de intracodificación o intercodificación. La memoria de datos de vídeo 69 y el DPB 82 pueden estar formados por cualquiera entre varios dispositivos de memoria, tales como memoria dinámica de acceso aleatorio (DRAM), incluyendo DRAM síncrona (SDRAM), RAM magneto-resistiva (MRAM), RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. La memoria de datos de vídeo 69 y el DPB 82 pueden ser proporcionados por el mismo dispositivo de memoria o por dispositivos de memoria independientes. En diversos ejemplos, la memoria de datos de vídeo 69 puede estar en un chip con otros componentes del decodificador de vídeo 30, o fuera de chip con respecto a esos componentes.

[0117] La memoria de datos de vídeo 69 puede recibir y almacenar datos de vídeo codificados (por ejemplo, unidades de NAL) de un flujo de bits. La unidad de decodificación por entropía 70 puede recibir datos de vídeo codificados (por ejemplo, unidades de NAL) desde la memoria de datos de vídeo 69 y puede analizar sintácticamente las unidades de NAL para decodificar elementos sintácticos. Durante el proceso de decodificación, el decodificador 30 de vídeo 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 decodificación de entropía 70 del decodificador de vídeo 30 decodifica por entropía el flujo de bits para generar coeficientes cuantizados, indicadores de modalidad de intra-predicción y otros elementos sintácticos.

[0118] La unidad de intra-predicción 74 puede generar datos de predicción para un bloque de vídeo del fragmento de vídeo actual basándose en una modalidad de intra-predicción señalizada y datos de bloques previamente decodificados de la trama o imagen actual. La unidad de compensación de movimiento 72 produce bloques predictivos para un bloque de vídeo del fragmento de vídeo actual basándose en los vectores de movimiento y otros elementos sintácticos recibidos desde la unidad de decodificación por entropía 70. Los bloques predictivos se pueden producir a partir de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. La unidad de compensación de movimiento 72 puede determinar la información de predicción para un bloque de vídeo usando una información de predicción, y puede producir los bloques predictivos para el bloque de vídeo actual que se está decodificando.

[0119] La unidad de cuantización inversa 76 puede cuantizar de manera inversa los coeficientes de transformación cuantizados, proporcionados en el flujo de bits y decodificados por la unidad de decodificación por entropía 70. La unidad de transformación inversa 78 aplica una transformación inversa, por ejemplo, una DCT inversa, una transformación inversa de enteros, o un proceso de transformación inversa conceptualmente similar, a los coeficientes de transformación, con el fin de generar bloques residuales en el dominio del píxel.

[0120] Después de que la unidad de compensación de movimiento 72 genera el bloque predictivo para el bloque de vídeo actual basándose en los vectores de movimiento y otros elementos sintácticos, el decodificador de vídeo 30 puede formar un bloque de vídeo decodificado sumando los bloques residuales de la unidad de transformación inversa 78 y los correspondientes bloques predictivos generados por la unidad de compensación de movimiento 72. El sumador 80 representa el componente o los componentes que realizan esta operación de suma. Los bloques de vídeo decodificados en una trama o imagen dada son a continuación almacenados en el DPB 82, que almacena imágenes de referencia usadas para la posterior compensación de movimiento. El DPB 82 almacena también vídeo decodificado para su presentación posterior en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la figura 1.

[0121] En algunos ejemplos, el decodificador de vídeo 30 recibe un flujo de transporte que incluye un primer descriptor, un segundo descriptor y una pluralidad de flujos elementales. El primer descriptor puede ser un descriptor para un punto de operación de los datos de vídeo. El segundo descriptor puede ser uno entre: un descriptor jerárquico

y un descriptor de extensión jerárquica. El decodificador de vídeo 30 puede decodificar un primer elemento sintáctico y un segundo elemento sintáctico en el primer descriptor. En al menos algunos de tales ejemplos, una lista de flujos elementales es una lista de flujos elementales del flujo de transporte que forman parte del punto de operación y el segundo descriptor tiene un valor de índice de capa jerárquica igual a un valor del segundo elemento sintáctico. Además, en dichos ejemplos, un primer valor del primer elemento sintáctico especifica que: se agregará un flujo elemental, indicado por el segundo elemento sintáctico, cuando no esté presente en la lista de flujos elementales, en la lista de flujos elementales, siendo el flujo elemental indicado por el segundo elemento sintáctico uno entre la pluralidad de flujos elementales, y se agregará un flujo elemental, indicado por un índice de capa integrada en el segundo descriptor, cuando no esté presente en la lista de flujos elementales, a la lista de flujos elementales, siendo el flujo elemental indicado por el índice de capa integrada en el segundo descriptor uno entre la pluralidad de flujos elementales. En respuesta a que la determinación del primer elemento sintáctico tiene un segundo valor diferente al primer valor, el decodificador de vídeo 30 puede agregar el flujo elemental indicado por el segundo elemento sintáctico, cuando no está presente en la lista de flujos elementales, en la lista de flujos elementales, pero no agregando el flujo elemental indicado por el índice de la capa integrada en el segundo descriptor en la lista de flujos elementales.

[0122] La figura 4 es un diagrama de flujo que ilustra una operación ejemplar de un dispositivo que genera un flujo de transporte, de acuerdo con una técnica de esta divulgación. Los diagramas de flujo de esta divulgación se proporcionan como ejemplos de operaciones. Otras operaciones de acuerdo con las técnicas de esta divulgación pueden incluir más, menos o diferentes acciones, o pueden realizar tales acciones en diferentes órdenes o en paralelo. En el ejemplo de la figura 4, el dispositivo puede ser un dispositivo de origen 12 u otro tipo de dispositivo para procesar datos de vídeo. El funcionamiento de la figura 4 puede implementar una solución a la segunda cuestión descrita anteriormente (es decir, el problema de no poder determinar los valores de ciertos elementos sintácticos para una capa derivada).

[0123] En el ejemplo de la figura 4, el dispositivo genera un primer descriptor (por ejemplo, un descriptor de punto de operación de la HEVC) (150). El primer descriptor es un descriptor de un punto de operación. Una lista de flujos elementales (por ejemplo, la *OperationPointESList[i]*) es una lista de flujos elementales del flujo de transporte que forman parte del punto de operación. Un dispositivo que recibe o procesa un flujo puede generar la lista de flujos elementales agregando flujos elementales a la lista de flujos elementales. Además, el dispositivo genera un segundo descriptor (152). El segundo descriptor es uno entre: un descriptor jerárquico y un descriptor de extensión jerárquica. El dispositivo puede incluir el primer descriptor y el segundo descriptor en un flujo de transporte (154). Por ejemplo, el dispositivo puede emitir una serie de bits que representan el primer descriptor y el segundo descriptor en el flujo de transporte.

[0124] Además, en el ejemplo de la figura 4, como parte de la generación del primer descriptor, el dispositivo incluye un primer elemento sintáctico (por ejemplo, *prepend_dependencias[i][j]*) en el primer descriptor (156). Además, como parte de la generación del primer descriptor, el dispositivo incluye un segundo elemento sintáctico (por ejemplo, *ES_reference[i][j]*) en el primer descriptor (158). El segundo descriptor tiene un valor de índice de capa jerárquica (por ejemplo, índice de capa jerárquica) igual a un valor del segundo elemento sintáctico. Cuando el dispositivo incluye un elemento sintáctico en un descriptor, el dispositivo puede almacenar un valor del elemento sintáctico en una memoria de manera que indique que el valor es parte del descriptor, por ejemplo, como parte de una estructura de datos.

[0125] En el ejemplo de la figura 4, un primer valor (por ejemplo, 1) del primer elemento sintáctico especifica que un flujo elemental indicado por el segundo elemento sintáctico, cuando no esté presente en la lista de flujos elementales, se agregará a la lista de flujos elementales. Además, el primer valor del primer elemento sintáctico especifica un flujo elemental indicado por un índice de capa integrada (por ejemplo, *hierarchy_embedded_layer_index* o *hierarchy_ext_embedded_layer_index*) en el segundo descriptor, cuando no esté presente en la lista de flujos elementales, se agregará a la lista de flujos elementales. Un segundo valor del primer elemento sintáctico especifica que el flujo elemental indicado por el segundo elemento sintáctico, cuando no esté presente en la lista de flujos elementales, se agregará a la lista de flujos elementales, pero no el flujo elemental indicado por el índice de capa integrada en el segundo descriptor.

[0126] La figura 5 es un diagrama de flujo que ilustra una operación ejemplar de un dispositivo que recibe un flujo de transporte, de acuerdo con una técnica de esta divulgación. En el ejemplo de la figura 5, el dispositivo puede ser un dispositivo de destino 14, un MANE u otro tipo de dispositivo para procesar datos de vídeo. La operación de la figura 5 puede implementar una solución a la segunda cuestión descrita anteriormente (es decir, el problema de no poder determinar los valores de ciertos elementos sintácticos para una capa obtenida).

[0127] En el ejemplo de la figura 5, el dispositivo recibe un flujo de transporte que incluye un primer descriptor (por ejemplo, un descriptor de punto de operación de la HEVC), un segundo descriptor y una pluralidad de flujos elementales (200). El primer descriptor es un descriptor para un punto de operación. El segundo descriptor es uno entre: un descriptor jerárquico y un descriptor de extensión jerárquica. En algunos ejemplos, la interfaz de entrada 28 recibe el flujo de transporte.

[0128] Además, el dispositivo puede decodificar un primer elemento sintáctico (por ejemplo, *prepend_dependencias[i][j]*) en el primer descriptor (202). En otras palabras, el dispositivo puede determinar un valor

del primer elemento sintáctico. Las formas ejemplares de decodificación de un elemento sintáctico pueden incluir analizar sintácticamente el elemento sintáctico a partir de una serie de bits, aplicar un algoritmo de decodificación por entropía u obtener de otro modo el valor del elemento sintáctico. Además, el dispositivo puede decodificar un segundo elemento sintáctico diferente (por ejemplo, ES_reference[i][j]) en el primer descriptor (204). En otras palabras, el dispositivo puede determinar un valor del segundo elemento sintáctico, por ejemplo, analizando sintácticamente el segundo elemento sintáctico a partir de los bits del primer descriptor. Una lista de flujos elementales (por ejemplo, la OperationPointESList[i]) es una lista de flujos elementales del flujo de transporte que forman parte del punto de operación. El segundo descriptor tiene un valor de índice de capa jerárquica (por ejemplo, hierarchy_layer_index) igual a un valor del segundo elemento sintáctico.

[0129] En el ejemplo de la figura 5, un primer valor (por ejemplo, 1) del primer elemento sintáctico especifica que un flujo elemental indicado por el segundo elemento sintáctico, cuando no esté presente en la lista de flujos elementales, se agregará a la lista de flujos elementales. El flujo elemental indicado por el segundo elemento sintáctico puede ser uno entre la pluralidad de flujos elementales recibidos por el dispositivo. Además, el primer valor del primer elemento sintáctico especifica un flujo elemental indicado por un índice de capa integrada (por ejemplo, hierarchy_embedded_layer_index o hierarchy_ext_embedded_layer_index) en el segundo descriptor, cuando no esté presente en la lista de flujos elementales, se agregará a la lista de flujos elementales. El flujo elemental indicado por el índice de capa integrada en el segundo descriptor puede ser uno entre la pluralidad de flujos elementales recibidos por el dispositivo. Un segundo valor (por ejemplo, 0) del primer elemento sintáctico especifica que el flujo elemental indicado por el segundo elemento sintáctico, cuando no esté presente en la lista de flujos elementales, se agregará a la lista de flujos elementales, pero no el flujo elemental indicado por el índice de capa integrada en el segundo descriptor.

[0130] Así, en algunos casos, tal como en el ejemplo de la figura 5, el dispositivo puede determinar que el primer elemento sintáctico tiene el segundo valor (206). En respuesta a que la determinación del primer elemento sintáctico tenga el segundo valor, el dispositivo puede agregar el flujo elemental indicado por el segundo elemento sintáctico, cuando no esté presente en la lista de flujos elementales, en la lista de flujos elementales, pero no agrega el flujo elemental, indicado por el índice de capa integrada en el segundo descriptor, en la lista de flujos elementales (208). Por ejemplo, un dispositivo puede agregar un flujo elemental a una lista almacenando una referencia al flujo elemental en una ubicación en la memoria, de una manera que indique que el flujo elemental está en la lista, tal como agregando la referencia al flujo elemental a un lista o formación enlazada.

[0131] Aunque no se ilustra en el ejemplo de la figura 5, el dispositivo puede descartar datos para capas que no corresponden al punto de operación. Por ejemplo, el dispositivo puede descartar paquetes de transporte de flujos elementales correspondientes a las capas que no corresponden al punto de operación. El dispositivo puede remitir los datos restantes (por ejemplo, paquetes de transporte de flujos elementales correspondientes a las capas del punto de operación) a otro dispositivo o un decodificador de vídeo, tal como el decodificador de vídeo 30.

[0132] El siguiente texto describe algunas formas ejemplares de implementar los aspectos descritos anteriormente. Los detalles de la implementación se describen en términos de cambios a DAM3. En los cambios de texto de estas implementaciones, el texto encerrado entre las etiquetas "<ins>" y "</ins>" se agrega o modifica (por ejemplo, <ins>texto insertado o modificado</ins>), mientras que las eliminaciones están encerradas entre las etiquetas "" y "" (por ejemplo, texto eliminado).

[0133] De acuerdo con una primera técnica ejemplar de implementación, la Tabla 1 y sus elementos sintácticos describen las modificaciones detalladas a DAM3 para los ejemplos primero, segundo y tercero para resolver la primera cuestión descrita anteriormente y el segundo ejemplo para resolver la segunda cuestión descrita anteriormente.

Tabla 1

Sintaxis	Nº de bits	Mnemónico
HEVC_operation_point_descriptor() { <ins>reserved</ins> num_ptl for (i = 0; i < num_ptl; i++, i++) { profile_tier_level_info[i] } operation_points_count for (i = 0; i < operation_points_count; i++) { target_ols[i] ES_count[i] } }	 <ins>2</ins> <ins>6</ins> 96 8 8 8	 <ins>bslbf</ins> uimsbf bslbf uimsbf uimsbf uimsbf

Sintaxis	Nº de bits	Mnemónico
<pre> HEVC_operation_point_descriptor() { for (j = 0; j < ES_count<ins>[i]</ins>; j++) { <dlt>output_layer_flag[i][j]</dlt> prepend_dependencies[i][j] ES_reference[i][j] <ins>reserved[i][j]</ins> <delete>ptl_ref_idx[i][j]</delete> } <ins>for (k = 0; k < NumESinOP[i]; k++) </pre>	<pre> <dlt>1</dlt> 1 6 <ins>1</ins> <dlt>8 </dlt> </pre>	<pre> <dlt>bslbf</dlt> bslbf uimsbf bslbf <dlt>uimsbf </dlt> </pre>
<pre> { necessary_layer_flag[i][k] output_layer_flag[i][k] <ins>ptl_ref_idx[i][k]</ins> } reserved avg_bit_rate_info_flag[i] max_bit_rate_info_flag[i] constant_frame_rate_info_idc[i] applicable_temporal_id[i] if (constant_frame_rate_info_idc[i] > 0) { reserved frame_rate_indicator[i] } if (avg_bit_rate_info_flag[i] == '1') { avg_bit_rate[i] } if (max_bit_rate_info_flag[i] == '1') { max_bit_rate[i] } } } </pre>	<pre> <ins>1</ins> <ins>1</ins> <ins>6</ins> 1 1 1 2 3 4 12 24 24 </pre>	<pre> <ins>bslbf</ins> <ins>bslbf</ins> <ins>uimsbf</ins> bslbf uimsbf bslbf uimsbf uimsbf bslbf uimsbf uimsbf </pre>

[0134] El siguiente texto indica modificaciones a la semántica de acuerdo con la primera técnica ejemplar de implementación. **num_ptl** - Este campo de <dlt> 8 bits </dlt><ins> 6 bits </ins> especifica el número de estructuras de perfil, grada y nivel señalizadas en este descriptor. <ins>Sea *OperationPointESList[i]* la lista de los ES que forman parte del i-ésimo punto de operación de la HEVC. </ins> **prepend_dependencies[i][j]** - Este indicador, si se fija en 1, <dlt> indica </dlt><ins> especifica que el ES indicado por *ES_reference[i][j]*, cuando no esté presente aún en la *OperationPointESList[i]*, se agregará a *OperationPointESList[i]* y <ins><dlt>que</dlt> el ES <ins>indicado</ins> por el elemento sintáctico *hierarchy_embedded_layer_index* en el descriptor jerárquico, o todos los ES <ins>indicados</ins> por el elemento sintáctico *hierarchy_ext_embedded_layer_index* en el descriptor de la extensión de jerarquía de la HEVC, con el valor del índice de capa jerárquica especificado por el siguiente elemento sintáctico *ES_reference[i][j] < ms >*, cuando aún no esté presente en la *OperationPointESList[i]*, <ins>será agregado <dlt>a los flujos elementales para el punto de operación de destino </dlt><ins> en la *ListaCapasPuntoOperación[i]* inmediatamente <ins>antes del ES señalado por la *ES_reference[i][j]* <ins> en orden ascendente del valor de su *hierarchy_embedded_layer_index* o *hierarchy_ext_embedded_layer_index* asociado. Cuando el valor de *prepend_dependencies[i][j]* es igual a 0, solo el ES indicado por *ES_reference[i][j]*, cuando aún no esté presente en la *OperationPointESList[i]*, se agregará a la *OperationPointESList[i]*. El ES indicado por la *ES_reference[i][m]* se colocará antes (es decir, con un índice inferior) en la *OperationPointESList[i]* que el ES indicado con la *ES_reference[i][n]* cuando *m* es menor que *n*. </ins><ins> NOTA: El orden de los ES en la *OperationPointESList[i]* debe estar en orden

ascendente de los valores de su *hierarchy_layer_index*. **ES_reference[i][j]** - Este campo de 6 bits indica el valor del índice de la capa jerárquica presente en el descriptor jerárquico o en el descriptor de la extensión jerárquica de la HEVC que identifica un flujo elemental. El valor de la *ES_reference[i][m]* y la *ES_reference[i][n]* para *m* no igual a *n* no será el mismo. Sea *NumESinOP[i]* el número de los ES en la *OperationPointESList[i]* después de que todos los ES que forman parte del *i*-ésimo punto de operación de la HEVC se hayan incluido en la *OperationPointESList[i]* (es decir, después de analizar sintácticamente *ES_reference[i][ES_count[i] - 1]*). **necessary_layer_flag[i][k]** - Este indicador, cuando se fija en '1', indica que el *k*-ésimo ES en la *OperationPointESList[i]* es una capa necesaria, según se define en 23008-2, del *i*-ésimo punto de operación. Este indicador igual a '0' indica que el *k*-ésimo ES en la *OperationPointESList[i]* no es una capa necesaria, según se define en 23008-2, del *i*-ésimo punto de operación. **output_layer_flag[i][k]** - Este indicador, cuando se fija en '1', indica que el *k*-ésimo ES en la capa de la *OperationPointESList[i]* es una capa de salida. De lo contrario, cuando se fija en '0', indica que el *k*-ésimo ES en la capa de la *OperationPointESList[i]* no es una capa de salida. Cuando el valor del *necessary_layer_flag[i][k]* es igual a 0, el valor del *output_layer_flag[i][k]* se ignorará. **ptl_ref_idx[i][k]** - Un campo de 8 bits que indica el índice *x* para el elemento *profile_tier_level_info[x]* de la *profile_tier_level_array* que se aplica al *k*-ésimo ES en la capa de la *OperationPointESList[i]* del *i*-ésimo punto de operación de la HEVC definido en este descriptor. Cuando el valor de *necessary_layer_flag[i][k]* es igual a 0, el valor de *ptl_ref_idx[i][k]* se ignorará.

[0135] La figura 6 es un diagrama de flujo que ilustra una operación ejemplar de un dispositivo para generar un flujo de transporte de acuerdo con una técnica ejemplar de esta divulgación. Por ejemplo, la operación de la figura 6 puede ser de acuerdo con la primera técnica ejemplar de implementación de esta divulgación expuesta anteriormente (es decir, la técnica descrita anteriormente con respecto a la Tabla 1). La operación de la figura 6 puede ser realizada por el dispositivo de origen 12 u otro dispositivo. La operación de la figura 6 puede ser una versión expandida de la operación de la figura 4.

[0136] En el ejemplo de la figura 6, el dispositivo puede generar un conjunto de segundos descriptores (250). Cada respectivo segundo descriptor del conjunto de segundos descriptores puede ser uno entre: un descriptor jerárquico y un descriptor de extensión jerárquica. El conjunto de segundos descriptores puede incluir el "segundo descriptor" descrito con respecto a la figura 4.

[0137] Además, el dispositivo puede generar un primer descriptor (por ejemplo, un descriptor de punto de operación de la HEVC) (252). Como parte de la generación del primer descriptor, el dispositivo puede incluir un elemento sintáctico de recuento de flujos elementales (por ejemplo, *ES_count*) en el primer descriptor (254). El elemento sintáctico de recuento de flujos elementales puede indicar un número de flujos elementales.

[0138] Además, como parte de la generación del primer descriptor, el dispositivo puede incluir un conjunto de primeros elementos sintácticos (por ejemplo, *prepend_dependencies[i][j]*) en el primer descriptor (256). Además, como parte de la generación del primer descriptor, el dispositivo puede incluir un conjunto de segundos elementos sintácticos (por ejemplo, *ES_reference[i][j]*) en el primer descriptor (258). El número de primeros elementos sintácticos en el conjunto de primeros elementos sintácticos y el número de segundos elementos sintácticos en el conjunto de segundos elementos sintácticos es igual a un valor del elemento sintáctico de recuento de flujo elemental. El conjunto de primeros elementos sintácticos puede incluir el "primer elemento sintáctico" mencionado con respecto a la figura 4. El conjunto de segundos elementos sintácticos puede incluir el "segundo elemento sintáctico" mencionado con respecto a la figura 4.

[0139] En algunos ejemplos, el conjunto de segundos elementos sintácticos está restringido de tal manera que no haya dos segundos elementos sintácticos del conjunto de segundos elementos sintácticos que tengan el mismo valor. En otras palabras, como se indica en los cambios a DAM3 en lo que antecede, el valor de *ES_reference[i][m]* y de *ES_reference[i][n]*, para *m* no igual a *n*, no será el mismo.

[0140] Para cada respectivo primer elemento sintáctico del conjunto de primeros elementos sintácticos, un respectivo segundo elemento sintáctico del conjunto de segundos elementos sintácticos corresponde al respectivo primer elemento sintáctico. Por ejemplo, un primer elemento sintáctico y un segundo elemento sintáctico con los mismos valores de *i* y *j* se corresponden entre sí. Un respectivo segundo descriptor del conjunto de segundos descriptores tiene un valor de índice de capa jerárquica (por ejemplo, *hierarchy_layer_id*) igual a un valor del respectivo segundo elemento sintáctico.

[0141] En el ejemplo de la figura 6, un primer valor (por ejemplo, 1) del respectivo primer elemento sintáctico especifica que un flujo elemental indicado por el respectivo segundo elemento sintáctico, cuando no esté presente en la lista de flujos elementales, se agregará a la lista de flujos elementales, y un flujo elemental indicado por un índice de capa integrada en el respectivo segundo descriptor, cuando no esté presente en la lista de flujos elementales del punto de operación, se agregará a la lista de flujos elementales. Un segundo valor (por ejemplo, 0) del respectivo primer elemento sintáctico especifica que el respectivo flujo elemental indicado por el respectivo segundo elemento

sintáctico, cuando no esté presente en la lista de flujos elementales, se agregará a la lista de flujos elementales, pero no el respectivo flujo elemental indicado por el índice de capa integrada en el respectivo segundo descriptor.

[0142] Además, en el ejemplo de la figura 6, el dispositivo puede incluir un conjunto de elementos sintácticos indicadores de capa necesaria (por ejemplo, `necessary_layer_flag[i][k]`) en el primer descriptor (260). Cada respectivo elemento sintáctico en el conjunto de elementos sintácticos indicadores de capa-necesaria puede corresponder a un respectivo flujo elemental en la lista de flujos elementales. Cada respectivo elemento sintáctico del conjunto de elementos sintácticos indicadores de capa-necesaria indica si el flujo elemental correspondiente al respectivo elemento sintáctico del conjunto de elementos sintácticos indicadores de capa-necesaria es una capa necesaria. Una capa necesaria puede definirse como una capa en un punto de operación de salida asociado a un conjunto de capas de salida, siendo la capa una capa de salida del conjunto de capas de salida, o una capa de referencia de una capa de salida del conjunto de capas de salida.

[0143] El dispositivo también puede incluir un conjunto de elementos sintácticos indicadores de capa de salida (por ejemplo, `output_layer_flag[i][k]`) en el primer descriptor (262). Cada respectivo elemento sintáctico en el conjunto de elementos sintácticos de salida puede corresponder a un respectivo flujo elemental en la lista de flujos elementales. Cada respectivo elemento sintáctico del conjunto de elementos sintácticos indicadores de capa de salida puede indicar si el flujo elemental, correspondiente al respectivo elemento sintáctico del conjunto de elementos sintácticos indicadores de capa de salida, es una capa de salida.

[0144] Además, el dispositivo puede incluir un conjunto de elementos sintácticos de índices de referencia de PTL (por ejemplo, `ptl_ref_idx[i][k]`) en el primer descriptor (264). Cada respectivo elemento sintáctico en el conjunto de elementos sintácticos de índices de referencia de PTL puede corresponder a un respectivo flujo elemental en la lista de flujos elementales. Cada respectivo elemento sintáctico del conjunto de elementos sintácticos de índices de referencia de PTL puede indicar un índice para determinar un perfil, grada o nivel del flujo elemental correspondiente al respectivo elemento sintáctico del conjunto de elementos sintácticos de índices de referencia de PTL.

[0145] En el ejemplo de la figura 6, el dispositivo puede incluir el primer descriptor y el segundo descriptor en un flujo de transporte (266). Por ejemplo, el dispositivo puede generar un conjunto de paquetes de transporte que contienen el primer descriptor y el segundo descriptor. El flujo de transporte puede incluir además uno o más flujos elementales. En otros ejemplos, una o más acciones de la figura 6 pueden ser omitidas.

[0146] La figura 7 es un diagrama de flujo que ilustra una operación ejemplar de un dispositivo que recibe un flujo de transporte de acuerdo con una técnica ejemplar de esta divulgación. Por ejemplo, la operación de la figura 7 puede ser congruente con la primera técnica ejemplar de implementación expuesta anteriormente. La operación de la figura 7 puede ser una versión expandida de la operación de la figura 5.

[0147] En el ejemplo de la figura 7, el dispositivo recibe un flujo de transporte que incluye un primer descriptor (por ejemplo, un descriptor de punto de operación de la HEVC), un conjunto de segundos descriptores y una pluralidad de flujos elementales (300). El conjunto de segundos descriptores puede incluir el "segundo descriptor" mencionado anteriormente con respecto a la figura 5. Cada respectivo segundo descriptor del conjunto de segundos descriptores puede ser uno entre: un descriptor jerárquico y un descriptor de extensión jerárquica.

[0148] Adicionalmente, en el ejemplo de la figura 7, el dispositivo puede decodificar un elemento sintáctico de recuento de flujo elemental (por ejemplo, `ES_count`, que puede ser mencionado en las reivindicaciones como un "tercer elemento sintáctico") en el primer descriptor. El elemento sintáctico de recuento de flujo elemental indica un número de flujos elementales.

[0149] Además, en el ejemplo de la figura 7, el dispositivo puede decodificar un conjunto de primeros elementos sintácticos (por ejemplo, `prepend_dependencias[i][j]`) en el primer descriptor (302). Además, en el ejemplo de la figura 7, el dispositivo puede decodificar un conjunto de segundos elementos sintácticos (por ejemplo, `ES_reference[i][j]`) en el primer descriptor (304). El número de primeros elementos sintácticos en el conjunto de primeros elementos sintácticos y el número de segundos elementos sintácticos en el conjunto de segundos elementos sintácticos es igual a un valor del tercer elemento sintáctico. El conjunto de primeros elementos sintácticos puede incluir el "primer elemento sintáctico" mencionado con respecto a la figura 5. El conjunto de segundos elementos sintácticos puede incluir el "segundo elemento sintáctico" mencionado con respecto a la figura 5.

[0150] En algunos ejemplos, el conjunto de segundos elementos sintácticos está restringido de tal manera que no haya dos segundos elementos sintácticos del conjunto de segundos elementos sintácticos que tengan el mismo valor. En otras palabras, como se indica en los cambios a DAM3 en lo que antecede, el valor de `ES_reference[i][m]` y de `ES_reference[i][n]`, para m no igual a n , no será el mismo.

[0151] Para cada respectivo primer elemento sintáctico del conjunto de primeros elementos sintácticos, un respectivo segundo elemento sintáctico del conjunto de segundos elementos sintácticos corresponde al respectivo primer elemento sintáctico. Por ejemplo, un primer elemento sintáctico y un segundo elemento sintáctico con los mismos valores de i y j se corresponden entre sí. Un respectivo segundo descriptor del conjunto de segundos

descriptores tiene un valor de índice de capa jerárquica (por ejemplo, `hierarchy_layer_id`) igual a un valor del respectivo segundo elemento sintáctico.

5 [0152] En el ejemplo de la figura 7, un primer valor (por ejemplo, 1) del respectivo primer elemento sintáctico especifica que un flujo elemental indicado por el respectivo segundo elemento sintáctico, cuando no esté presente en la lista de flujos elementales, se agregará a la lista de flujos elementales, y un flujo elemental, indicado por un índice de capa integrada en el respectivo segundo descriptor, cuando no esté presente en la lista de flujos elementales del punto de operación, se agregará a la lista de flujos elementales. El flujo elemental indicado por el respectivo segundo elemento sintáctico puede ser uno entre la pluralidad de flujos elementales. Un segundo valor (por ejemplo, 0) del respectivo primer elemento sintáctico especifica que el respectivo flujo elemental, indicado por el respectivo segundo elemento sintáctico, cuando no esté presente en la lista de flujos elementales, se agregará a la lista de flujos elementales, pero no el respectivo flujo elemental indicado por el índice de capa integrada en el respectivo segundo descriptor. El flujo elemental indicado por el índice de capa integrada en el respectivo segundo descriptor puede ser uno entre la pluralidad de flujos elementales.

15 [0153] Para cada respectivo primer elemento sintáctico del conjunto de primeros elementos sintácticos, se incluye un conjunto de uno o más índices de capa integrada en el respectivo segundo descriptor. Además, en el ejemplo de la figura 7, basándose en el respectivo primer elemento sintáctico que tiene el primer valor, el dispositivo puede agregar, en orden ascendente del valor del conjunto de índices de capa integrada, los flujos elementales indicados por el conjunto de índices de capa integrada en la lista de flujos elementales inmediatamente antes del flujo elemental indicado por el respectivo segundo elemento sintáctico (308). En otras palabras, en respuesta a la determinación de que el respectivo primer elemento sintáctico tiene el segundo valor, el dispositivo puede agregar el respectivo flujo elemental indicado por el respectivo segundo elemento sintáctico, cuando no esté presente en la lista de flujos elementales, en la lista de flujos elementales, pero no agrega el respectivo flujo elemental, indicado por el índice de capa integrada en el respectivo segundo descriptor, en la lista de flujos elementales.

20 [0154] En el ejemplo de la figura 7, el dispositivo puede decodificar un conjunto de elementos sintácticos indicadores de capa-necesaria (por ejemplo, `necessary_layer_flag[i][k]`) en el primer descriptor (310). Cada respectivo elemento sintáctico en el conjunto de elementos sintácticos indicadores de capa-necesaria puede corresponder a un respectivo flujo elemental en la lista de flujos elementales. Cada respectivo elemento sintáctico del conjunto de elementos sintácticos de bandera de capa-necesaria puede indicar si el flujo elemental correspondiente al respectivo elemento sintáctico del conjunto de elementos sintácticos indicadores de capa-necesaria es una capa necesaria. Como se ha descrito anteriormente, una capa necesaria se define como una capa en un punto de operación de salida, asociado con un conjunto de capas de salida, siendo la capa una capa de salida del conjunto de capas de salida, o una capa de referencia de una capa de salida del conjunto de capas de salida.

30 [0155] Además, en el ejemplo de la figura 7, el dispositivo puede decodificar un conjunto de elementos sintácticos indicadores de capa de salida (por ejemplo, `output_layer_flag[i][k]`) en el primer descriptor (312). Cada respectivo elemento sintáctico en el conjunto de elementos sintácticos indicadores de capa de salida puede corresponder a un respectivo flujo elemental en la lista de flujos elementales. Cada respectivo elemento sintáctico del conjunto de elementos sintácticos indicadores de capa de salida indica si el flujo elemental, correspondiente al respectivo elemento sintáctico del conjunto de elementos sintácticos indicadores de capa de salida, es una capa de salida.

35 [0156] Además, el dispositivo puede decodificar un conjunto de elementos sintácticos de índices de referencia de PTL en el primer descriptor (314). Cada respectivo elemento sintáctico en el conjunto de elementos sintácticos de índices de referencia de PTL puede corresponder a un respectivo flujo elemental en la lista de flujos elementales. Cada respectivo elemento sintáctico del conjunto de elementos sintácticos de índices de referencia de PTL puede indicar un índice para determinar un perfil, grada o nivel del flujo elemental correspondiente al respectivo elemento sintáctico del conjunto de elementos sintácticos de índices de referencia de PTL.

40 [0157] En el ejemplo de la figura 7, el dispositivo puede descartar datos para capas que no forman parte del punto de operación (316). En otras palabras, el dispositivo puede descartar datos para capas que no corresponden al punto de operación. En otros ejemplos, una o más acciones de la figura 7 pueden ser omitidas.

45 [0158] De acuerdo con una segunda técnica ejemplar de implementación, la Tabla 2 y sus elementos sintácticos describen las modificaciones detalladas a DAM3 para la tercera técnica ejemplar para resolver la segunda cuestión descrita anteriormente.

Tabla 2

Sintaxis	Nº de bits	Mnemónico
<code>HEVC_operation_point_descriptor() {</code> <code><ins>reserved</ins></code> <code>num_ptl</code>	<code><ins>2</ins></code> <code><ins>6</ins></code>	<code><ins>bslbf</ins></code> <code>uimsbf</code>

Sintaxis	Nº de bits	Mnemónico
<pre> HEVC_operation_point_descriptor() { for (i = 0; i < num_ptl; i++, i++) { profile_tier_level_info[i] } operation_points_count for (i = 0; i < operation_pointcount; i++) { target_ols[i] ES_count[i] for (j = 0; j < ES_count<ins>[i]</ins>; j++) { output_layer_flag[i][j] <ins>ptl_ref_idx[i][j]</ins> es_present_flag[i][j] if (es_present_flag[i][j]) { reserved[i][j]</ins> </pre>	<p>96</p> <p>8</p> <p>8</p> <p>8</p> <p>1</p> <p><ins>6</ins></p> <p><ins>1</ins></p> <p>1</p>	<p>bslbf</p> <p>uimsbf</p> <p>uimsbf</p> <p>uimsbf</p> <p>bslbf</p> <p><ins>uimsbf</ins></p> <p>bslbf</p> <p>bslbf</p>
<pre> prepend_dependencies[i][j] ES_reference[i][j] } <dlt>ptl_ref_idx[i][j]</dlt> } } reserved avg_bit_rate_info_flag[i] max_bit_rate_info_flag[i] constant_frame_rate_info_idc[i] applicable_temporal_id[i] if (constant_frame_rate_info_idc[i] > 0) { reserved frame_rate_indicator[i] } if (avg_bit_rate_info_flag[i] == '1') { avg_bit_rate[i] } if (max_bit_rate_info_flag[i] == '1') { max_bit_rate[i] } } } </pre>	<p>1</p> <p>6</p> <p><dlt>8</dlt></p> <p>1</p> <p>1</p> <p>1</p> <p>2</p> <p>3</p> <p>4</p> <p>12</p> <p>24</p> <p>24</p>	<p>bslbf</p> <p>uimsbf</p> <p><</p> <p>dlt>uimsbf</dlt></p> <p>bslbf</p> <p>bslbf</p> <p>bslbf</p> <p>uimsbf</p> <p>uimsbf</p> <p>bslbf</p> <p>uimsbf</p> <p>uimsbf</p>

5 **[0159]** El siguiente texto indica modificaciones a la semántica de acuerdo con la segunda técnica ejemplar de implementación. El <ins> **es_present_flag[i][j]** especifica si ES_reference[i][j] y prepend_dependencies[i][j] están presentes para la j-ésima capa del i-ésimo punto de operación. Cuando el es_present_flag[i][j] es igual a 0, el valor de ES_reference[i][j] se obtiene basándose en información en el descriptor jerárquico o el descriptor de extensión jerárquica de la capa señalada previamente en el i-ésimo punto de operación con los valores del es_present_flag igual a 1 y los valores de prepend_dependencies igual a 1. Cuando el es_present_flag[i][j] es igual a 0, se obtiene el valor de prepend_dependencies[i][j] es 0. El valor del es_present_flag[i][0] no será igual a 0. </ins>

10 **prepend_dependencies[i][j]** - Este indicador, si se fija en 1, indica que el ES señalado por el elemento sintáctico *hierarchy_embedded_layer_index* en el descriptor de la jerarquía, o todos los ES señalizados por el elemento sintáctico *hierarchy_ext_embedded_layer_index* en el descriptor de extensión jerárquica de la HEVC, con el valor del

índice de capa jerárquica especificado por el siguiente elemento sintáctico *ES_reference[i][j]*, se agregará a la lista de flujos elementales para el punto de operación de destino *<dlt>antes de </dlt>* *<ins>* después *</ins>* del ES señalado por la *ES_reference[i][j]*. *<ins>* NOTA: La lista posterior de flujos elementales para el punto de operación de destino está rellena; se reordenará en el orden ascendente de su valor de índice de capa jerárquica. *</ins>* Obsérvese que las soluciones para los problemas de la capa necesaria y la posibilidad de una inclusión duplicada del ES en la lista de los ES para una operación de la HEVC también se pueden aplicar a esta alternativa.

5

[0160] De acuerdo con una tercera técnica ejemplar de implementación, la Tabla 3 y sus elementos sintácticos describen las modificaciones detalladas para DAM3 para la cuarta técnica ejemplar para resolver la segunda cuestión descrita anteriormente.

10

Tabla 3

Sintaxis	Nº de bits	Mnemónico
<pre> HEVC_operation_point_descriptor () { num_ptl for (i = 0; i < num_ptl; i++, i++) { profile_tier_level_info[i] } operation_points_count for (i = 0; i < operation_points_count; i++) </pre>	<p>8</p> <p>96</p> <p>8</p>	<p>uimsbf</p> <p>bslbf</p> <p>uimsbf</p>
<pre> { target_ols[i] <ins>reserved[i]</ins> ES_count[i] <ins>default_output_ref_layer[i] </ins> for (j = 0; j < ES_count; j++) { output_layer_flag[i][j] prepend_dependencies[i][j] ES_reference[i][j] ptl_ref_idx[i][j] } reserved avg_bit_rate_info_flag[i] max_bit_rate_info_flag[i] constant_frame_rate_info_idc[i] applicable_temporal_id[i] if (constant_frame_rate_info_idc[i] > 0){ reserved frame_rate_indicator[i] } if (avg_bit_rate_info_flag[i] == '1') { avg_bit_rate[i] } if (max_bit_rate_info_flag[i] == '1') { max_bit_rate[i] } } </pre>	<p>8</p> <p><ins>1</ins></p> <p><ins>6</ins></p> <p><ins>1</ins></p> <p>1</p> <p>1</p> <p>6</p> <p>8</p> <p>1</p> <p>1</p> <p>1</p> <p>2</p> <p>3</p> <p>4</p> <p>12</p> <p>24</p> <p>24</p>	<p>uimsbf</p> <p><ins>bslbf</ins></p> <p>uimsbf</p> <p><ins>bslbf</ins></p> <p>bslbf</p> <p>bslbf</p> <p>uimsbf</p> <p>uimsbf</p> <p>bslbf</p> <p>uimsbf</p> <p>uimsbf</p> <p>uimsbf</p> <p>uimsbf</p> <p>uimsbf</p> <p>uimsbf</p>

Sintaxis	Nº de bits	Mnemónico
HEVC_operation_point_descriptor() { } }		

[0161] El siguiente texto indica modificaciones a la semántica de acuerdo con la tercera técnica ejemplar de implementación.

5 **prepend_dependencies[i][j]** - Este indicador, si se fija en 1, indica que el ES señalado por el elemento sintáctico *hierarchy_embedded_layer_index* en el descriptor jerárquico, o la totalidad de los ES señalizados por el elemento sintáctico *hierarchy_ext_embedded_layer_index* en el descriptor de extensión de jerarquía de la HEVC, con el valor del índice de capa jerárquica especificado por el siguiente elemento sintáctico *ES_reference[i][j]*, se agregarán a la lista de flujos elementales para el punto de operación de destino antes del ES señalado por la *ES_reference[i][j]*.

10 <ins> Cuando *prepend_dependencies[i][j]* es igual a 1, para cada flujo elemental *esA* que esté señalado por el elemento sintáctico *hierarchy_embedded_layer_index* en el descriptor jerárquico, o todos los flujos elementales señalizados por el elemento sintáctico *hierarchy_ext_embedded_layer_index* en el descriptor de extensión de jerarquía de la HEVC, con el valor de índice de capa jerárquica especificado por el siguiente elemento sintáctico *ES_reference[i][j]*, vale lo siguiente:

- 15 • Si la *default_output_ref_layer[i]* es igual a 0, la capa asociada al *esA* no es una capa de salida del *i*-ésimo punto de operación de la HEVC; de lo contrario, la capa asociada al *esA* es una capa de salida del *i*-ésimo punto de operación de la HEVC.
- 20 • Habrá al menos un punto de operación de la *opX* de la HEVC con índice entre 0 e *i* - 1 que también contuvo el *esA*. Si hay más de un candidato para *opX*, se usará el que tenga el índice más alto. El índice del elemento *profile_tier_level_info[x]* de la *profile_tier_level_array* que se aplica al *esA* en el *i*-ésimo punto de operación de la HEVC es el conjunto igual al índice del elemento *profile_tier_level_info[x]* de la *profile_tier_level_array* que se aplica al *esA* en *opX*. </ins>

[0162] Además, para evitar complicaciones innecesarias, *prepend_dependencies* se pueden señalar en el nivel del punto de operación, en lugar del nivel de la capa. Las soluciones para los problemas de la capa necesaria y la posibilidad de la inclusión duplicada de ES en la lista de los ES para una operación de la HEVC también se pueden aplicar a esta alternativa.

35 **prepend_dependencies[i]** - Este indicador, si se fija en 1, indica que el ES señalado por el elemento sintáctico *hierarchy_embedded_layer_index* en el descriptor jerárquico, o todos los ES señalizados por el elemento sintáctico *hierarchy_ext_embedded_layer_index* en el descriptor de extensión jerárquica de la HEVC, con el valor del índice de capa jerárquica especificado por el siguiente elemento sintáctico *referenciaES[i][j]*, se agregará a la lista de flujos elementales para el punto de operación de destino antes del ES señalado por la *ES_reference[i][j]*

40 <ins>, para cada flujo elemental indicado por la *referencia ES[i][j]* del *i*-ésimo punto de operación. </ins><ins> Cuando *prepend_dependencies[i]* es igual a 1, para cada flujo elemental *esA* que esté señalado por el elemento sintáctico *hierarchy_embedded_layer_index* en el descriptor jerárquico, o por todos los flujos elementales señalizados por el elemento sintáctico *hierarchy_ext_embedded_layer_index* en el descriptor de extensión jerárquica de la HEVC, con el valor de índice de capa jerárquica especificado por el siguiente elemento sintáctico *ES_reference[i][j]*, vale lo siguiente:

- 45 • Si la *default_output_ref_layer[i]* es igual a 0, la capa asociada al *esA* no es una capa de salida del *i*-ésimo punto de operación de la HEVC; de lo contrario, la capa asociada al *esA* es una capa de salida del *i*-ésimo punto de operación de la HEVC.
- 50 • Habrá al menos un punto de operación de la HEVC, *opX*, con índice entre 0 e *i* - 1 que también contuviera el *esA*. Si hay más de un candidato para *opX*, se usará el que tenga el índice más alto. El índice del elemento *profile_tier_level_info [x]* de la *profile_tier_level_array* que se aplica al *esA* en el *i*-ésimo punto de operación de la HEVC es el conjunto igual al índice del elemento *profile_tier_level_info [x]* de la *profile_tier_level_array* que se aplica al *esA* en *opX*. </ins>

55 [0163] Además de la cuestión en DAM3 descrita anteriormente, cuando un flujo de bits de la HEVC en capas contiene una o más capas auxiliares, no hay ningún mecanismo para describir la mejora auxiliar en el descriptor jerárquico o el descriptor de extensión jerárquica. Una imagen auxiliar es una imagen que no tiene ningún efecto normativo en el proceso de decodificación de las imágenes primarias, y con un valor de *nuh_layer_id* tal que *AuxId[nuh_layer_id]* sea mayor que 0. Una capa auxiliar es una capa que contiene imágenes auxiliares. Para resolver

este problema, se sugiere incluir una descripción para el tipo de jerarquía auxiliar en la Tabla 2-50 y la mejora auxiliar en la Tabla 2-103.

5 **[0164]** Para la gestión de capas auxiliares, la HEVC en capas da soporte a la presencia de capas auxiliares en el flujo de bits. Una capa auxiliar puede contener, por ejemplo, representación de canal alfa o profundidad. El texto actual de ISO / IEC 13818-1: 201x / DAM 3 Transporte de HEVC en capas no tiene el mecanismo para describir las capas auxiliares. Llevar capas auxiliares de SHVC / MV-HEVC en un flujo elemental con un tipo de flujo específico (por ejemplo, 0x1E, es decir, el flujo de video auxiliar según se define en ISO / IEC 23002-3) no es una solución suficiente porque todavía no hay ninguna manera de describir la jerarquía de este flujo elemental y otros flujos elementales, especialmente el flujo al que se asocia la capa auxiliar en un programa. Además, el modelo actual de almacén temporal L-HEVC no gestiona tipos de flujo distintos a 0x27~0x2A, por lo que esta solución (utilizando un tipo de flujo específico) puede requerir modificaciones adicionales al modelo de almacén temporal.

15 **[0165]** Para superar el problema anterior, la divulgación describe la inclusión del tipo de jerarquía auxiliar en la Tabla 2-50 y la inclusión de la mejora auxiliar en la Tabla 2-103undeciens, a continuación. Con estas modificaciones, no han ninguna necesidad adicional de asignar un tipo de flujo específico para las capas auxiliares, ninguna necesidad adicional de actualizar la modalidad de almacén temporal, y una capa auxiliar puede transportarse en un flujo elemental con un tipo de flujo en el rango de 0x27 ~ 0x2A.

20 **[0166]** El cambio de texto propuesto para la Tabla 2-50 es el siguiente:
Reemplace en la Tabla 2-50 la descripción de los valores 8<ins>, 10</ins> y 15, de la siguiente manera:

Tabla 2-50 - Valores del campo hierarchy_type

Valor	Descripción
8	Ajustabilidad a escala combinada o sub-partición de MV-HEVC
<ins>10</ins>	<ins>Auxiliar</ins>
15	Capa de base o sub-flujo de bits de la vista de base de la MVC o sub-flujo de bits de vídeo de la AVC del sub-flujo temporal de bits de vídeo de la MVC o la HEVC o sub-partición de base de la HEVC.

25 Propuesta de cambio de texto a la tabla 2-103undeciens:

Tabla 2-103undeciens - Semántica de bits de dimensión de extensión

Índice para los bits	Descripción
0	Mejora de vista múltiple
1	Ajustabilidad a escala espacial, incluyendo la SNR
2	Mejora de la profundidad
3	Mejora temporal
<ins>4</ins>	<ins> Mejora auxiliar</ins>
5~15	Reservado

30 **[0167]** Los siguientes párrafos proporcionan ejemplos adicionales de esta divulgación.

35 **[0168]** En algunos ejemplos, el decodificador de vídeo 30 puede configurarse para recibir un primer elemento sintáctico (por ejemplo, `prepend_dependencias[i][j]`) para un punto de operación, en donde un primer valor del elemento sintáctico especifica que un flujo elemental indicado por un segundo elemento sintáctico (por ejemplo, `ES_reference[i][j]`), cuando aún no esté presente en una lista de puntos de operación (por ejemplo, la `OperationPointESList[i]`), se agregará a la lista de puntos de operación, y en donde un segundo valor del elemento sintáctico especifica que solo el flujo elemental indicado por el segundo elemento sintáctico, cuando aún no esté presente en la lista de operaciones, se agregará a la lista de puntos de operación, y decodificar el punto de operación en función del primer elemento sintáctico recibido.

40 **[0169]** En algunos ejemplos, el decodificador de vídeo 30 puede configurarse para realizar un primer bucle para decodificar múltiples instancias de un primer elemento sintáctico (por ejemplo, `ES_reference[i][j]`) que indica el valor de índice de capa jerárquica presente en el descriptor jerárquico o el descriptor de extensión jerárquica de la HEVC, que identifica un flujo elemental de un punto de operación, y / o decodificar múltiples instancias de un segundo elemento sintáctico (por ejemplo, `prepend_dependencias[i][j]`) que indica a partir de qué estructuras han de incluirse flujos elementales en una lista de flujos elementales para el punto de operación, como parte del rendimiento del primer bucle, realizando un segundo bucle para decodificar múltiples instancias de un tercer elemento sintáctico (por ejemplo, `output_layer_flag[i][k]`) que indica qué flujo elemental del punto de operación es una capa de salida y / o decodificar

múltiples instancias de un cuarto elemento sintáctico (por ejemplo, `ptl_ref_idx[i][k]`) que indica un índice para determinar un perfil, grada o nivel de un flujo elemental del punto de operación, y decodificar el punto de operación en función de uno o más de los elementos sintácticos primero, segundo, tercero y cuarto.

5 **[0170]** En algunos ejemplos, el codificador de vídeo 20 o un dispositivo intermedio (por ejemplo, un MANE) puede configurarse para implementar las técnicas ejemplares descritas en esta divulgación. Por ejemplo, el codificador de vídeo 20 o el dispositivo intermedio (o una combinación de los dos) puede configurarse para determinar flujos o capas elementales de un punto de operación, determinar un indicador (por ejemplo, `necessary_layer_flag[i][k]`, descrito a continuación) para cada flujo o capa elemental en el punto de operación, en donde el indicador indica si un respectivo
10 flujo o capa elemental es un flujo, o capa, necesario, y produce un flujo de bits que incluye los flujos elementales y los indicadores respectivos.

[0171] En un ejemplo, el codificador de vídeo 20 o el dispositivo intermedio (o una combinación de los dos) puede configurarse para determinar un primer elemento sintáctico (por ejemplo, `prepend_dependencies[i][j]`) para un punto de operación, en donde un primer valor del elemento sintáctico especifica que un flujo elemental indicado por un segundo elemento sintáctico (por ejemplo, `ES_reference[i][j]`), cuando aún no esté presente en una lista de puntos de operación (por ejemplo, la `OperationPointESList[i]`), se agregará a la lista de puntos de operación, y en donde un segundo valor del elemento sintáctico especifica que solo el flujo elemental indicado por el segundo elemento sintáctico, cuando aún no esté presente en la lista de operaciones, se agregará a la lista de puntos de operación, y producirá un flujo de bits que incluya información del punto de operación que incluye el primer elemento sintáctico determinado.

[0172] En un ejemplo, el codificador de vídeo 20 o el dispositivo intermedio (o una combinación de los dos) puede configurarse para realizar un primer bucle para determinar múltiples instancias de un primer elemento sintáctico (por ejemplo, `ES_reference[i][j]`) que indica el valor de índice de capa jerárquica presente en el descriptor jerárquico o en el descriptor de extensión jerárquica de la HEVC, que identifica un flujo elemental de un punto de operación, y / o determinar múltiples instancias de un segundo elemento sintáctico (por ejemplo, `prepend_dependencies[i][j]`) que indica de cuáles estructuras han de incluirse los flujos elementales en una lista de flujos elementales para el punto de operación, como parte del rendimiento del primer bucle, realizando un segundo bucle para determinar múltiples instancias de un tercer elemento sintáctico (por ejemplo, `output_layer_flag[i][k]`) que indica qué flujo elemental del punto de operación es una capa de salida y / o determinar múltiples instancias de un cuarto elemento sintáctico (por ejemplo, `ptl_ref_idx[i][k]`) que indica un índice para determinar un perfil, grada o nivel de un flujo elemental del punto de operación, y producir un flujo de bits que incluye uno o más de los elementos sintácticos primero, segundo, tercero y cuarto para el punto de operación.

[0173] Debe reconocerse que, según el 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 hilos, el procesamiento de interrupciones o múltiples procesadores, en lugar de secuencialmente.

[0174] En uno o más ejemplos, las funciones descritas pueden implementarse en hardware, software, firmware o cualquier combinación de los mismos. Si se implementan en software, las funciones pueden almacenarse en, y transmitirse por, un medio legible por ordenador, como una o más instrucciones o código, y ejecutarse mediante una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que correspondan a un medio tangible tal como medios de almacenamiento de datos o medios de comunicación que incluyan 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 legibles por ordenador que sean no transitorios o (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible al que se pueda acceder desde uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para la implementación de las técnicas descritas en esta divulgación. Un producto de programa informático puede incluir un medio legible por ordenador.

[0175] A modo de ejemplo, y no de manera limitativa, tales medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento de disco óptico, almacenamiento de disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que pueda usarse para almacenar código de programa deseado en forma de instrucciones o estructuras de datos y al que pueda accederse mediante un ordenador. Además, cualquier conexión recibe debidamente la denominación de medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde una sede de la Red, un servidor u otro origen remoto 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, están orientados a medios de almacenamiento tangibles no transitorios. El término disco, como se usa en el presente documento, incluye un disco compacto (CD), un disco láser, un disco óptico, un disco versátil digital (DVD), un disco flexible y un disco Blu-ray, donde algunos discos normalmente emiten datos magnéticamente, mientras que otros discos emiten datos ópticamente con láseres. Las combinaciones de lo anterior también deberían incluirse dentro del alcance de los medios legibles por ordenador.

[0176] 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 de compuertas programables in situ (FPGA) u otros circuitos lógicos integrados o discretos equivalentes. En consecuencia, el término "procesador", como se usa en el presente documento, se puede referir a cualquiera de las estructuras anteriores 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 se puede proporcionar dentro de módulos de hardware y/o software dedicados, configurados para la codificación y la decodificación, o incorporados en un códec combinado. También, las técnicas se podrían implementar totalmente en uno o más circuitos o elementos lógicos.

[0177] Las técnicas de la presente divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, incluidos un equipo manual inalámbrico, un circuito integrado (CI) o un conjunto de CI (por ejemplo, un conjunto de chips). Diversos componentes, módulos o unidades se describen en esta divulgación para enfatizar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no requieren necesariamente su realización mediante diferentes unidades de hardware. En cambio, como se ha descrito anteriormente, diversas unidades se pueden combinar en una unidad de hardware de códec, o ser proporcionadas por un grupo de unidades de hardware interoperativas, incluyendo uno o más procesadores, como se ha descrito anteriormente, juntamente con software y/o firmware adecuados.

[0178] Se han descrito diversos ejemplos. Estos y otros ejemplos están dentro del alcance de las siguientes reivindicaciones.

REIVINDICACIONES

1. Un procedimiento para procesamiento de datos de vídeo, el procedimiento que comprende:

5 recibir (200; 300) un flujo de transporte que incluye un primer descriptor, un segundo descriptor y una pluralidad de flujos elementales, siendo el primer descriptor un descriptor para un punto de operación de los datos de vídeo, siendo el segundo descriptor uno entre: un descriptor de jerarquía o un descriptor de extensión jerárquica;

10 decodificar (202, 204; 304, 306) un primer elemento sintáctico y un segundo elemento sintáctico en el primer descriptor, en donde:

15 una lista de flujos elementales es una lista de flujos elementales del flujo de transporte que forman parte del punto de operación,

el segundo descriptor tiene un valor de índice de capa jerárquica igual a un valor del segundo elemento sintáctico,

20 un primer valor del primer elemento sintáctico especifica que:

un flujo elemental indicado por el segundo elemento sintáctico, cuando no esté presente en la lista de flujos elementales, se agregará a la lista de flujos elementales, siendo el flujo elemental indicado por el segundo elemento sintáctico uno entre la pluralidad de flujos elementales, y

25 un flujo elemental indicado por un índice de capa integrada en el segundo descriptor, cuando no esté presente en la lista de flujos elementales, se agregará a la lista de flujos elementales, siendo el flujo elemental indicado por el índice de capa integrada en el segundo descriptor uno entre la pluralidad de flujos elementales, y

30 en respuesta a la determinación (206) de que el primer elemento sintáctico tiene un segundo valor diferente al primer valor, agregar (208; 308) el flujo elemental indicado por el segundo elemento sintáctico, cuando no esté presente en la lista de flujos elementales, a la lista de flujos elementales, pero no agregar el flujo elemental, indicado por el índice de capa integrada en el segundo descriptor, en la lista de flujos elementales.

35 2. El procedimiento según la reivindicación 1, que comprende además:

recibir (300) un conjunto de segundos descriptores en el flujo de transporte, incluyendo el conjunto de segundos descriptores el segundo descriptor, siendo cada respectivo segundo descriptor del conjunto de segundos descriptores uno entre: un descriptor jerárquico o un descriptor de extensión jerárquica;

40 decodificar (302) un tercer elemento sintáctico en el primer descriptor, indicando el tercer elemento sintáctico un número de flujos elementales;

45 decodificar (304) un conjunto de primeros elementos sintácticos en el primer descriptor; y

decodificar (306) un conjunto de segundos elementos sintácticos en el primer descriptor, en donde:

50 el número de primeros elementos sintácticos en el conjunto de primeros elementos sintácticos y el número de segundos elementos sintácticos en el conjunto de segundos elementos sintácticos es igual a un valor del tercer elemento sintáctico,

el conjunto de primeros elementos sintácticos incluye el primer elemento sintáctico,

55 el conjunto de segundos elementos sintácticos incluye el segundo elemento sintáctico,

para cada respectivo primer elemento sintáctico del conjunto de primeros elementos sintácticos:

60 un respectivo segundo elemento sintáctico del conjunto de segundos elementos sintácticos corresponde al respectivo primer elemento sintáctico,

un respectivo segundo descriptor del conjunto de segundos descriptores tiene un valor de índice de capa jerárquica igual a un valor del respectivo segundo elemento sintáctico,

65 el primer valor del respectivo primer elemento sintáctico especifica que:

un flujo elemental indicado por el respectivo segundo elemento sintáctico, cuando no esté presente en la lista de flujos elementales, se agregará a la lista de flujos elementales, siendo el flujo elemental indicado por el respectivo segundo elemento sintáctico uno entre la pluralidad de flujos elementales, y

5 un flujo elemental indicado por un índice de capa integrada en el respectivo segundo descriptor, cuando no esté presente en la lista de flujos elementales del punto de operación, se agregará a la lista de flujos elementales, siendo el flujo elemental indicado por el índice de capa integrada en el respectivo segundo descriptor uno entre la pluralidad de flujos elementales, y

10 en respuesta a la determinación de que el respectivo primer elemento sintáctico tiene el segundo valor, agregar (308) el respectivo flujo elemental, indicado por el respectivo segundo elemento sintáctico, cuando no esté presente en la lista de flujos elementales, a la lista de flujos elementales, pero sin agregar el respectivo flujo elemental, indicado por el índice de capa integrada en el respectivo segundo descriptor, en la lista de flujos elementales.

15 **3.** El procedimiento según la reivindicación 2, que comprende además:

20 para cada respectivo primer elemento sintáctico del conjunto de primeros elementos sintácticos:

un conjunto de uno o más índices de capa integrada se incluye en el respectivo segundo descriptor, y

25 basándose en el respectivo primer elemento sintáctico que tiene el primer valor, agregar, en orden ascendente del valor del conjunto de índices de capa integrada, flujos elementales, indicados por el conjunto de índices de capa integrada, en la lista de flujos elementales inmediatamente antes del flujo elemental indicado por el respectivo segundo elemento sintáctico.

4. El procedimiento según la reivindicación 1, que comprende además:

30 decodificar (310) un conjunto de elementos sintácticos en el primer descriptor, correspondiendo cada respectivo elemento sintáctico en el conjunto de elementos sintácticos a un respectivo flujo elemental en la lista de flujos elementales, indicando cada respectivo elemento sintáctico del conjunto de elementos sintácticos si el flujo elemental correspondiente al respectivo elemento sintáctico del conjunto de elementos sintácticos es una capa necesaria, en donde una capa necesaria se define como una capa en un punto de operación de salida, asociado a un conjunto de capas de salida, siendo la capa una capa de salida del conjunto de capas de salida, o una capa de referencia de una capa de salida del conjunto de capas de salida;

35 o

40 decodificar (312) un primer conjunto de elementos sintácticos en el primer descriptor, correspondiendo cada respectivo elemento sintáctico en el primer conjunto de elementos sintácticos a un respectivo flujo elemental en la lista de flujos elementales, indicando cada respectivo elemento sintáctico del primer conjunto de elementos sintácticos si el flujo elemental correspondiente al respectivo elemento sintáctico del primer conjunto de elementos sintácticos es una capa de salida; y

45 decodificar (314) un segundo conjunto de elementos sintácticos en el primer descriptor, correspondiendo cada respectivo elemento sintáctico en el segundo conjunto de elementos sintácticos a un respectivo flujo elemental en la lista de flujos elementales, indicando cada respectivo elemento sintáctico del segundo conjunto de elementos sintácticos un índice para determinar un perfil, grada o nivel del flujo elemental correspondiente al respectivo elemento sintáctico del segundo conjunto de elementos sintácticos;

50 o

55 recibir (300) un conjunto de segundos descriptores en el flujo de transporte, incluyendo el conjunto de segundos descriptores el segundo descriptor, siendo cada respectivo segundo descriptor del conjunto de segundos descriptores uno entre: un descriptor jerárquico y un descriptor de extensión jerárquica; y

60 decodificar (306) un conjunto de segundos elementos sintácticos en el primer descriptor, en donde:

el conjunto de segundos elementos sintácticos incluye el segundo elemento sintáctico,

65 indicando cada respectivo segundo elemento sintáctico del conjunto de segundos elementos sintácticos un valor de índice de capa jerárquica presente en un descriptor del conjunto de segundos descriptores, y

el conjunto de segundos elementos sintácticos está restringido de modo tal que no haya dos segundos elementos sintácticos del conjunto de segundos elementos sintácticos que tengan el mismo valor;

o

5

descartar (316) datos para capas que no forman parte del punto de operación.

5. Un dispositivo (14) para procesar datos de vídeo, el dispositivo que comprende:

10

una interfaz de entrada (28) configurada para recibir información que incluye un flujo de transporte que incluye un primer descriptor, un segundo descriptor y una pluralidad de flujos elementales, siendo el primer descriptor un descriptor para un punto de operación de los datos de vídeo, siendo el segundo descriptor uno entre: un descriptor jerárquico o un descriptor de extensión jerárquica; y

15

uno o más procesadores (29, 30) configurados para:

decodificar un primer elemento sintáctico y un segundo elemento sintáctico en el primer descriptor, en donde:

20

una lista de flujos elementales es una lista de flujos elementales del flujo de transporte que forman parte del punto de operación,

el segundo descriptor tiene un valor de índice de capa jerárquica igual a un valor del segundo elemento sintáctico,

25

un primer valor del primer elemento sintáctico especifica que:

30

un flujo elemental indicado por el segundo elemento sintáctico, cuando no esté presente en la lista de flujos elementales, se agregará a la lista de flujos elementales, siendo el flujo elemental indicado por el segundo elemento sintáctico uno entre la pluralidad de flujos elementales, y

35

un flujo elemental indicado por un índice de capa integrada en el segundo descriptor, cuando no esté presente en la lista de flujos elementales, se agregará a la lista de flujos elementales, siendo el flujo elemental indicado por el índice de capa integrada en el segundo descriptor uno entre la pluralidad de flujos elementales, y

40

en respuesta a la determinación de que el primer elemento sintáctico tiene un segundo valor diferente al primer valor, agregar el flujo elemental indicado por el segundo elemento sintáctico, cuando no esté presente en la lista de flujos elementales, en la lista de flujos elementales, pero no agregar el flujo elemental indicado por el índice de capa integrada en el segundo descriptor en la lista de flujos elementales.

6. El dispositivo (14) según la reivindicación 5, en el que los uno o más procesadores (29, 30) están configurados para:

45

recibir un conjunto de segundos descriptores en el flujo de transporte, incluyendo el conjunto de segundos descriptores el segundo descriptor, siendo cada respectivo segundo descriptor del conjunto de segundos descriptores uno entre: un descriptor jerárquico o un descriptor de extensión jerárquica;

50

decodificar un tercer elemento sintáctico en el primer descriptor, indicando el tercer elemento sintáctico un número de flujos elementales;

decodificar un conjunto de primeros elementos sintácticos en el primer descriptor; y

55

decodificar un conjunto de segundos elementos sintácticos en el primer descriptor, en donde:

60

el número de primeros elementos sintácticos en el conjunto de primeros elementos sintácticos y el número de segundos elementos sintácticos en el conjunto de segundos elementos sintácticos es igual a un valor del tercer elemento sintáctico,

65

el conjunto de primeros elementos sintácticos incluye el primer elemento sintáctico,

el conjunto de segundos elementos sintácticos incluye el segundo elemento sintáctico,

para cada respectivo primer elemento sintáctico del conjunto de primeros elementos sintácticos:

un respectivo segundo elemento sintáctico del conjunto de segundos elementos sintácticos corresponde al respectivo primer elemento sintáctico,

5 un respectivo segundo descriptor del conjunto de segundos descriptores tiene un valor de índice de capa jerárquica igual a un valor del respectivo segundo elemento sintáctico,

el primer valor del respectivo primer elemento sintáctico especifica que:

10 un flujo elemental indicado por el respectivo segundo elemento sintáctico, cuando no esté presente en la lista de flujos elementales, se agregará a la lista de flujos elementales, siendo el flujo elemental indicado por el respectivo segundo elemento sintáctico uno entre la pluralidad de flujos elementales, y

15 un flujo elemental indicado por un índice de capa integrada en el respectivo segundo descriptor, cuando no esté presente en la lista de flujos elementales del punto de operación, se agregará a la lista de flujos elementales, siendo el flujo elemental indicado por el índice de capa integrada en el respectivo segundo descriptor uno entre la pluralidad de flujos elementales, y

20 en respuesta a la determinación de que el primer elemento sintáctico respectivo tiene el segundo valor, los uno o más procesadores están configurados para agregar el respectivo flujo elemental, indicado por el respectivo segundo elemento sintáctico, cuando no esté presente en la lista de flujos elementales, en la lista de flujos elementales, pero no agregar el respectivo flujo elemental, indicado por el índice de capa integrada en el respectivo segundo descriptor, en la lista de flujos elementales.

25 7. El dispositivo (14) según la reivindicación 6, en el que, para cada respectivo primer elemento sintáctico del conjunto de primeros elementos sintácticos, se incluye un conjunto de uno o más índices de capa integrada en el respectivo segundo descriptor, y los uno o más procesadores (29, 30) están configurados de tal manera que, en función del respectivo primer elemento sintáctico que tiene el primer valor, los uno o más procesadores agregan, en orden ascendente del valor del conjunto de índices de capa integrada, flujos elementales indicados por el conjunto de índices de capa integrada en la lista de flujos elementales inmediatamente antes del flujo elemental indicado por el respectivo segundo elemento sintáctico.

30 8. El dispositivo (14) según la reivindicación 5, en el que los uno o más procesadores (29, 30) están configurados para:

35 decodificar un conjunto de elementos sintácticos en el primer descriptor, correspondiendo cada respectivo elemento sintáctico en el conjunto de elementos sintácticos a un respectivo flujo elemental en la lista de flujos elementales, indicando cada respectivo elemento sintáctico del conjunto de elementos sintácticos si el flujo elemental corresponde al respectivo elemento sintáctico del conjunto de elementos sintácticos es una capa necesaria, en donde una capa necesaria se define como una capa en un punto de operación de salida, asociado a un conjunto de capas de salida, siendo la capa una capa de salida del conjunto de capas de salida, o una capa de referencia de una capa de salida del conjunto de capas de salida; o

40 decodificar un primer conjunto de elementos sintácticos en el primer descriptor, correspondiendo cada respectivo elemento sintáctico en el primer conjunto de elementos sintácticos a un respectivo flujo elemental en la lista de flujos elementales, indicando cada respectivo elemento sintáctico del primer conjunto de elementos sintácticos si el flujo elemental correspondiente al respectivo elemento sintáctico del primer conjunto de elementos sintácticos es una capa de salida; y

45 decodificar un segundo conjunto de elementos sintácticos en el primer descriptor, correspondiendo cada respectivo elemento sintáctico en el segundo conjunto de elementos sintácticos a un respectivo flujo elemental en la lista de flujos elementales, indicando cada respectivo elemento sintáctico del segundo conjunto de elementos sintácticos un índice para determinar un perfil, grada o nivel del flujo elemental correspondiente al respectivo elemento sintáctico del segundo conjunto de elementos sintácticos;

50 o

55 recibir un conjunto de segundos descriptores en el flujo de transporte, incluyendo el conjunto de segundos descriptores el segundo descriptor, siendo cada respectivo segundo descriptor del conjunto de segundos descriptores uno entre: un descriptor jerárquico o un descriptor de extensión jerárquica; y

60 decodificar un conjunto de segundos elementos sintácticos en el primer descriptor, en donde:

65 el conjunto de segundos elementos sintácticos incluye el segundo elemento sintáctico,

indicando cada respectivo segundo elemento sintáctico del conjunto de segundos elementos sintácticos un valor de índice de capa jerárquica presente en un descriptor del conjunto de segundos descriptores, y

5 el conjunto de segundos elementos sintácticos está restringido de modo tal que no haya dos segundos elementos sintácticos del conjunto de segundos elementos sintácticos que tengan el mismo valor; o

descartar datos para las capas que no forman parte del punto de operación.

10 **9.** El dispositivo (14) de cualquiera de las reivindicaciones 5 a 8, que comprende además un dispositivo de visualización (32) para exhibir datos de vídeo decodificados.

15 **10.** El dispositivo (14) de cualquiera de las reivindicaciones 5 a 9, en el que el dispositivo (14) es un dispositivo de comunicación inalámbrica.

11. Un dispositivo (12) para procesar datos de vídeo, el dispositivo que comprende: uno o más procesadores (20, 21) configurados para:

20 generar un primer descriptor, siendo el primer descriptor un descriptor de un punto de operación de los datos de vídeo;

generar un segundo descriptor, siendo el segundo descriptor uno entre: un descriptor jerárquico o un descriptor de extensión jerárquica;

25 en donde los uno o más procesadores están configurados de tal manera que, como parte de la generación del primer descriptor, los uno o más procesadores:

incluyen un primer elemento sintáctico y un segundo elemento sintáctico en el primer descriptor, en donde:

30 una lista de flujos elementales es una lista de flujos elementales que forman parte del punto de operación,

35 el segundo descriptor tiene un valor de índice de capa jerárquica igual a un valor del segundo elemento sintáctico,

un primer valor del primer elemento sintáctico especifica que:

40 un flujo elemental indicado por el segundo elemento sintáctico, cuando no esté presente en la lista de flujos elementales, se agregará a la lista de flujos elementales, y

un flujo elemental indicado por un índice de capa integrada en el segundo descriptor, cuando no esté presente en la lista de flujos elementales, se agregará a la lista de flujos elementales, y

45 un segundo valor del primer elemento sintáctico especifica que el flujo elemental indicado por el segundo elemento sintáctico, cuando no esté presente en la lista de flujos elementales, se agregará a la lista de flujos elementales, pero no el flujo elemental indicado por el índice de capa integrada en el segundo descriptor; e

50 incluir el primer descriptor y el segundo descriptor en un flujo de transporte; y una interfaz de salida (22) configurada para generar una versión codificada de los datos de vídeo.

12. El dispositivo (12) según la reivindicación 11, en el que:

55 los uno o más procesadores (20, 21) están configurados para generar un conjunto de segundos descriptores, incluyendo el conjunto de segundos descriptores el segundo descriptor, siendo cada respectivo segundo descriptor del conjunto de segundos descriptores uno entre: un descriptor jerárquico o un descriptor de extensión jerárquica;

60 los uno o más procesadores se configuran de tal manera que, como parte de la generación del primer descriptor, los uno o más procesadores:

incluyen un tercer elemento sintáctico en el primer descriptor, indicando el tercer elemento sintáctico un número de flujos elementales;

65 incluyen un conjunto de primeros elementos sintácticos en el primer descriptor; e

incluyen un conjunto de segundos elementos sintácticos en el primer descriptor, en donde:

5 el número de primeros elementos sintácticos en el conjunto de primeros elementos sintácticos y el número de segundos elementos sintácticos en el conjunto de segundos elementos sintácticos es igual a un valor del tercer elemento sintáctico,

el conjunto de primeros elementos sintácticos incluye el primer elemento sintáctico,

10 el conjunto de segundos elementos sintácticos incluye el segundo elemento sintáctico,

para cada respectivo primer elemento sintáctico del conjunto de primeros elementos sintácticos:

15 un respectivo segundo elemento sintáctico del conjunto de segundos elementos sintácticos corresponde al respectivo primer elemento sintáctico,

un respectivo segundo descriptor del conjunto de segundos descriptores tiene un valor de índice de capa jerárquica igual a un valor del respectivo segundo elemento sintáctico,

20 el primer valor del respectivo primer elemento sintáctico especifica que:

un flujo elemental indicado por el respectivo segundo elemento sintáctico, cuando no esté presente en la lista de flujos elementales, se agregará a la lista de flujos elementales, y

25 un flujo elemental indicado por un índice de capa integrada en el respectivo segundo descriptor, cuando no esté presente en la lista de flujos elementales del punto de operación, se agregará a la lista de flujos elementales, y

30 el segundo valor del respectivo primer elemento sintáctico especifica que el respectivo flujo elemental indicado por el respectivo segundo elemento sintáctico, cuando no esté presente en la lista de flujos elementales, se agregará a la lista de flujos elementales, pero no el respectivo flujo elemental indicado por el índice de capa integrada en el respectivo segundo descriptor.

35 **13.** El dispositivo (12) de la reivindicación 12, en el que los uno o más procesadores (20, 21) están configurados de tal manera que, como parte de la generación del primer descriptor, los uno o más procesadores:

40 incluyen un conjunto de elementos sintácticos en el primer descriptor, correspondiendo cada respectivo elemento sintáctico en el conjunto de elementos sintácticos a un respectivo flujo elemental en la lista de flujos elementales, indicando cada respectivo elemento sintáctico del conjunto de elementos sintácticos si el flujo elemental correspondiente al respectivo elemento sintáctico del conjunto de elementos sintácticos es una capa necesaria, en donde una capa necesaria se define como una capa en un punto de operación de salida asociado a un conjunto de capas de salida, siendo la capa una capa de salida del conjunto de capas de salida, o una capa de referencia de una capa de salida del conjunto de capas de salida.

45 **14.** El dispositivo (12) de la reivindicación 11, en el que:

los uno o más procesadores (20, 21) están configurados de tal manera que, como parte de la generación del primer descriptor, los uno o más procesadores:

50 incluyen un primer conjunto de elementos sintácticos en el primer descriptor, correspondiendo cada respectivo elemento sintáctico en el primer conjunto de elementos sintácticos a un respectivo flujo elemental en la lista de flujos elementales, indicando cada respectivo elemento sintáctico del primer conjunto de elementos sintácticos si el flujo elemental correspondiente al respectivo elemento sintáctico del primer conjunto de elementos sintácticos es una capa de salida; e

55 incluyen un segundo conjunto de elementos sintácticos en el primer descriptor, correspondiendo cada respectivo elemento sintáctico en el segundo conjunto de elementos sintácticos a un respectivo flujo elemental en la lista de flujos elementales, indicando cada respectivo elemento sintáctico del segundo conjunto de elementos sintácticos un índice para determinar un perfil, grada o nivel del flujo elemental correspondiente al respectivo elemento sintáctico del segundo conjunto de elementos sintácticos;

o

65 los uno o más procesadores (20, 21) están configurados además para incluir un conjunto de segundos descriptores en el flujo de transporte, incluyendo el conjunto de segundos descriptores el segundo descriptor,

siendo cada uno de los respectivos segundos descriptores del conjunto de segundos descriptores uno entre: un descriptor jerárquico o un descriptor de extensión jerárquica, y

5 los uno o más procesadores se configuran de tal manera que, como parte de la generación del primer descriptor, los uno o más procesadores:

incluyen un conjunto de segundos elementos sintácticos en el primer descriptor, en donde:

10 el conjunto de segundos elementos sintácticos incluye el segundo elemento sintáctico,

indicando cada respectivo segundo elemento sintáctico del conjunto de segundos elementos sintácticos un valor de índice de capa jerárquica presente en un descriptor del conjunto de segundos descriptores, y

15 el conjunto de segundos elementos sintácticos está restringido de manera tal que no haya dos segundos elementos sintácticos del conjunto de segundos elementos sintácticos que tengan el mismo valor.

15. El dispositivo (14) de cualquiera de las reivindicaciones 11 a 14, en el que el dispositivo (14) es un dispositivo de comunicación inalámbrica.

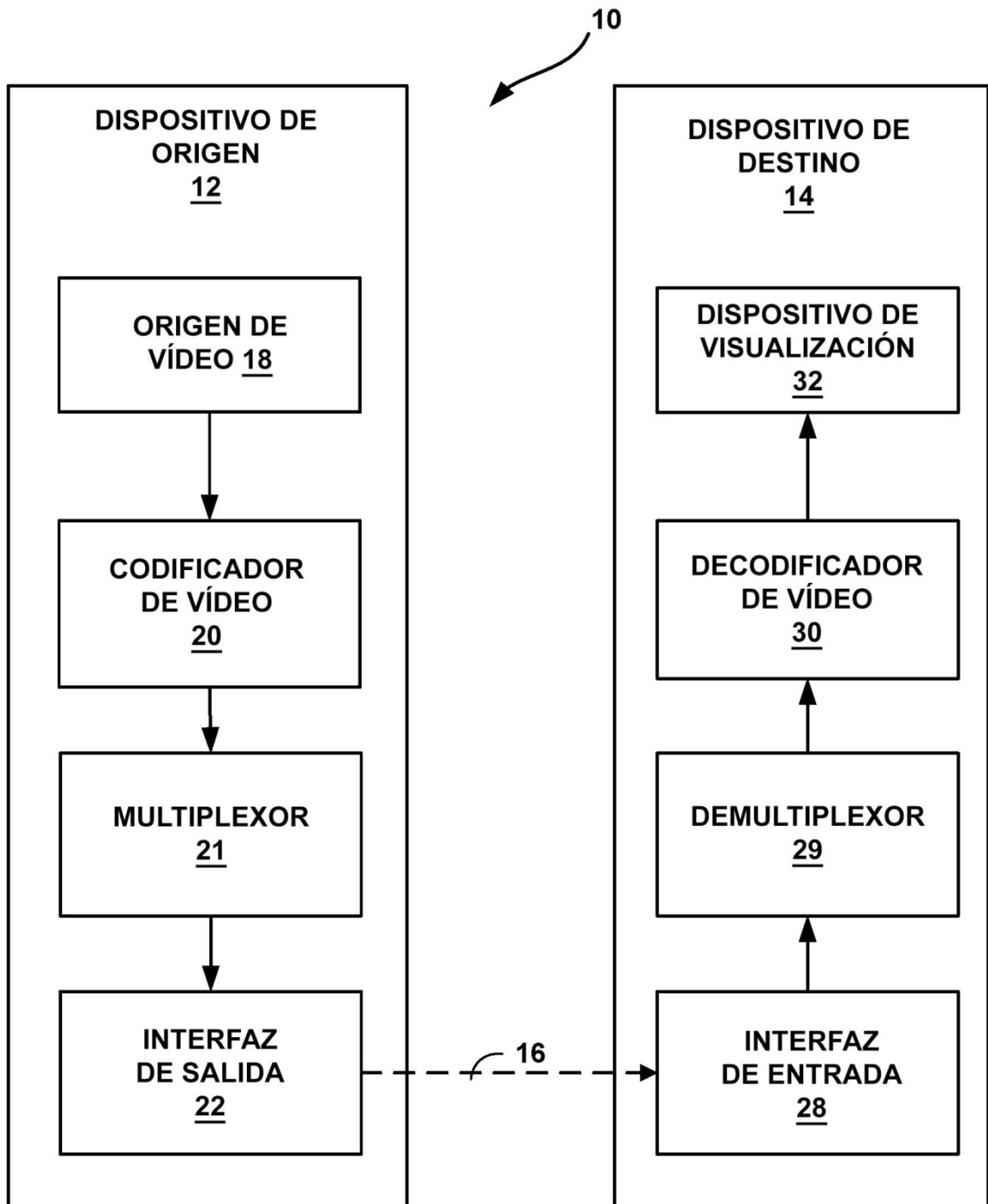


FIG. 1

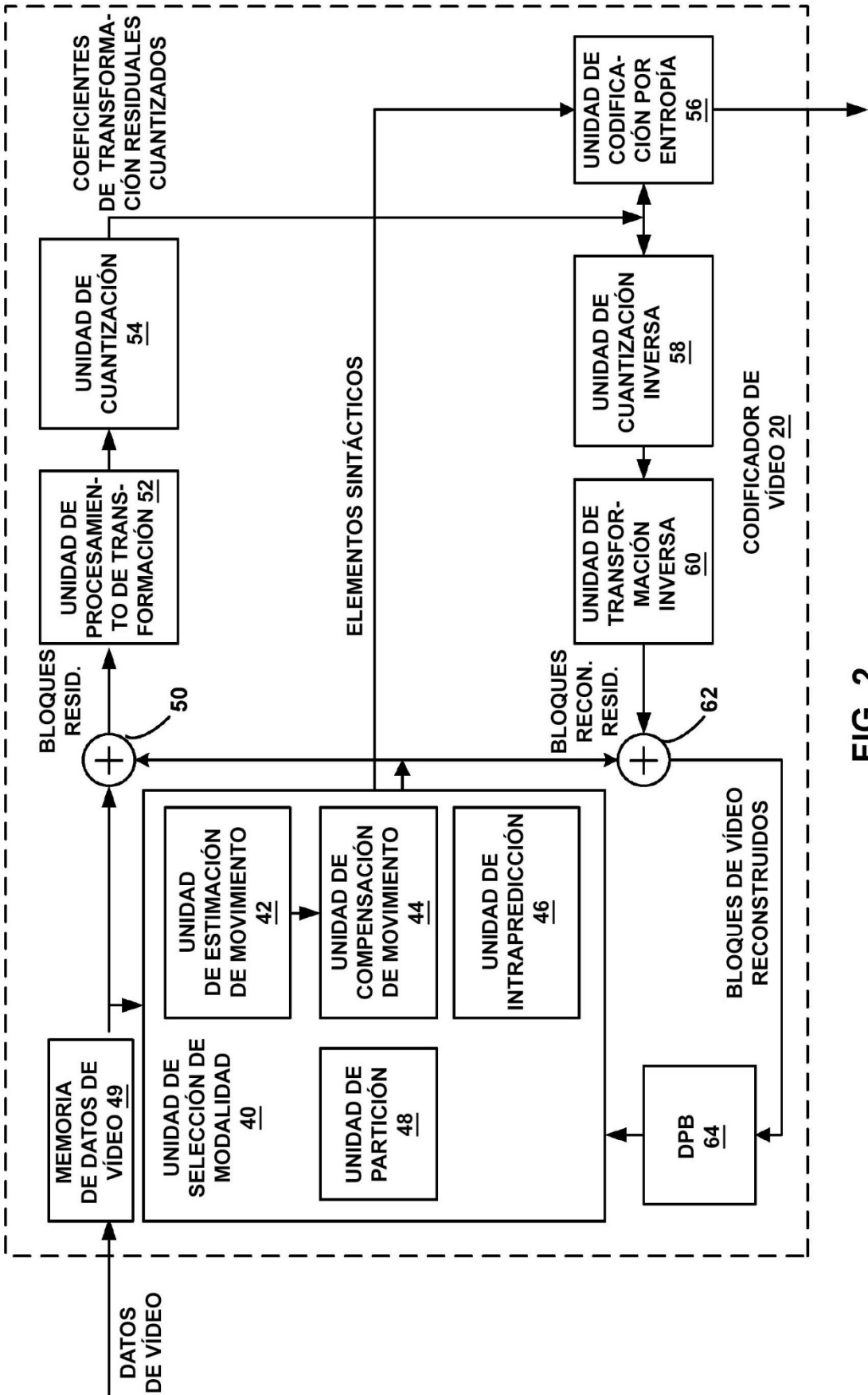


FIG. 2

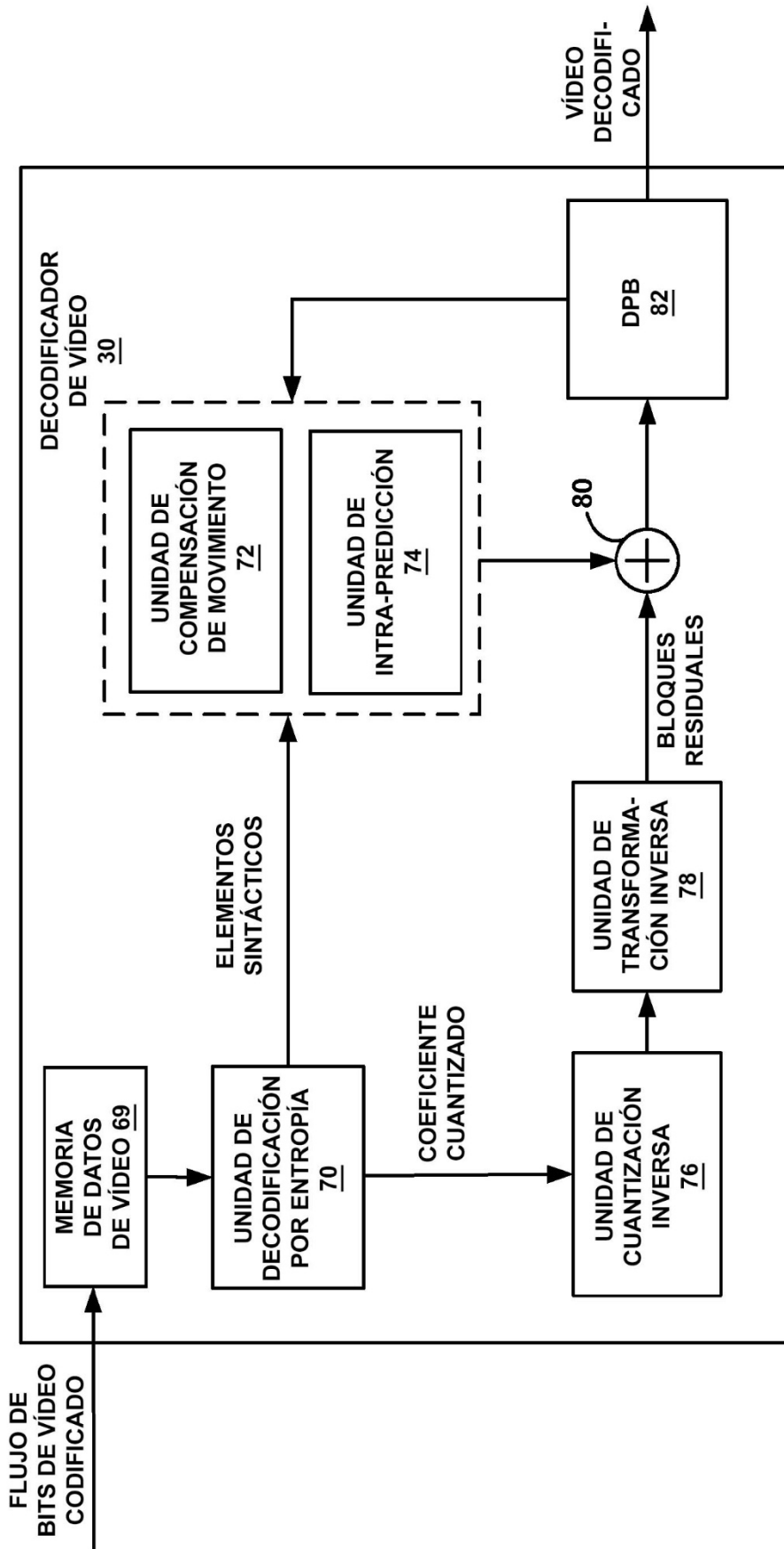


FIG. 3

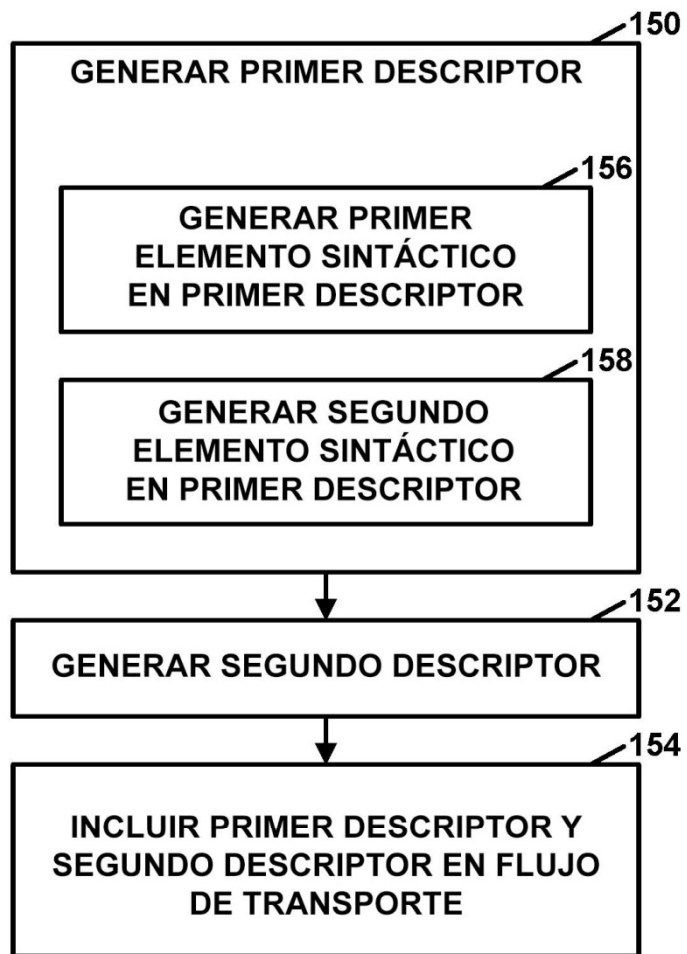


FIG. 4

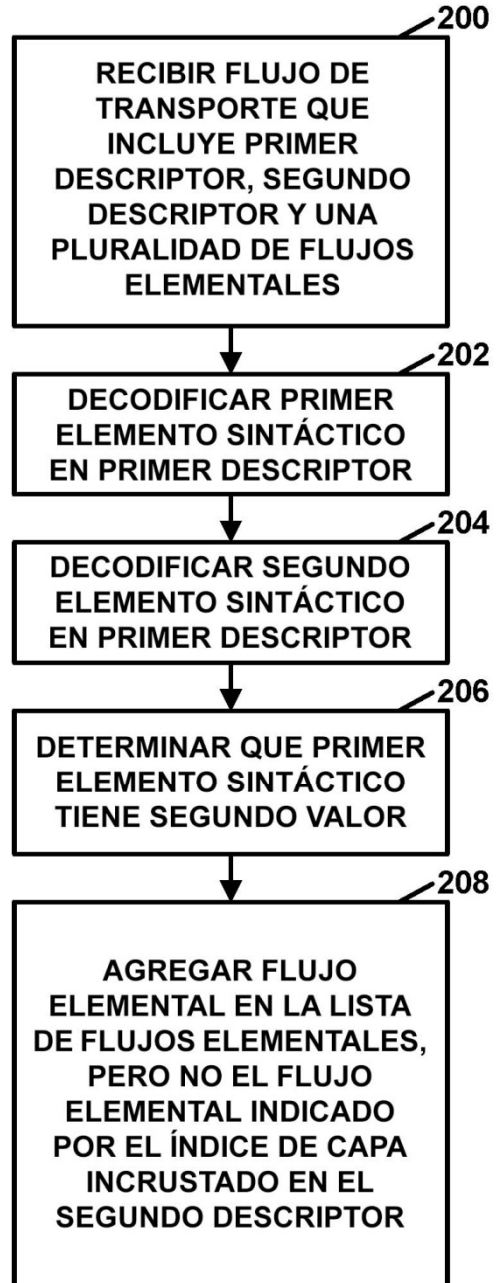


FIG. 5

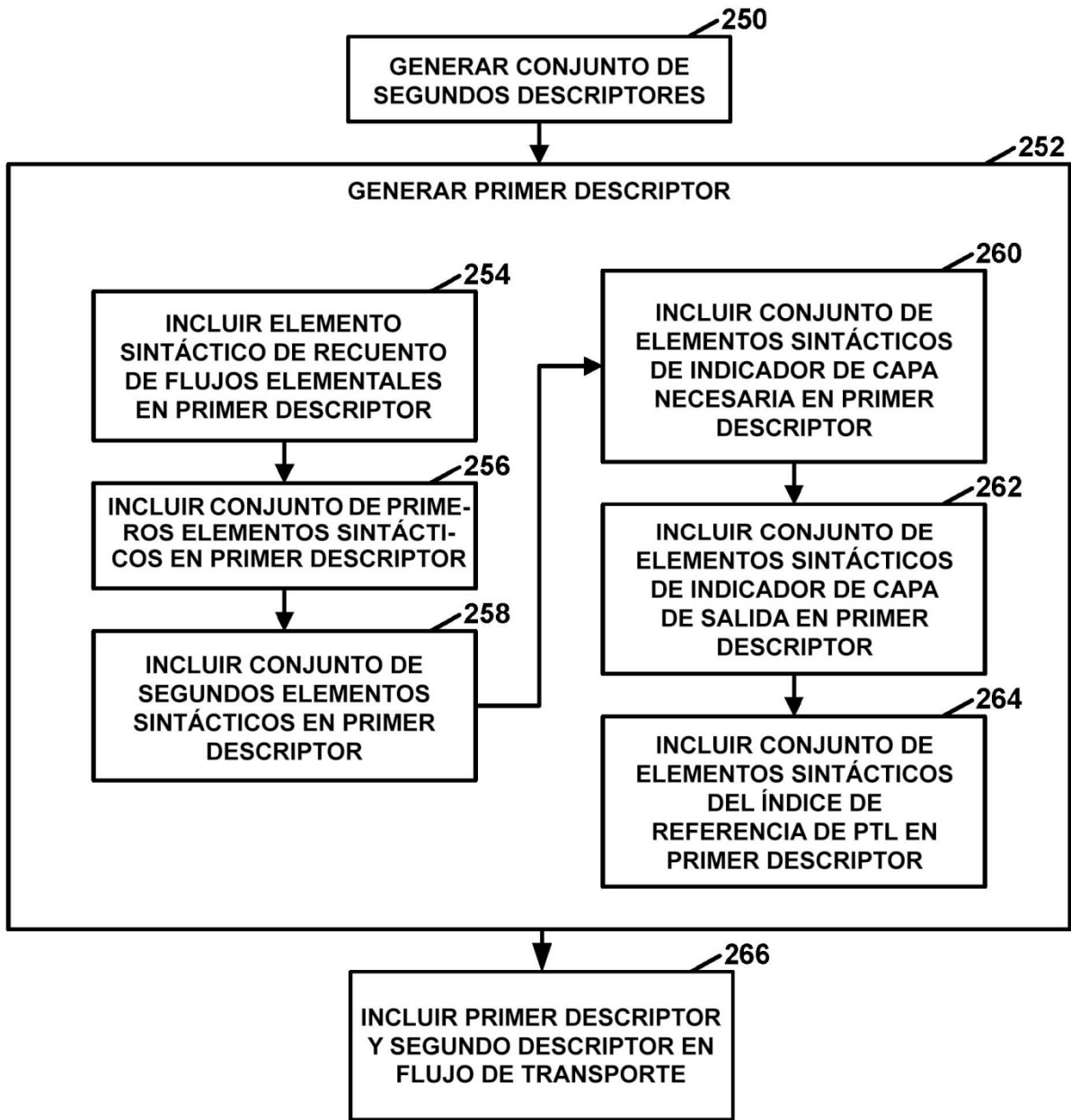


FIG. 6

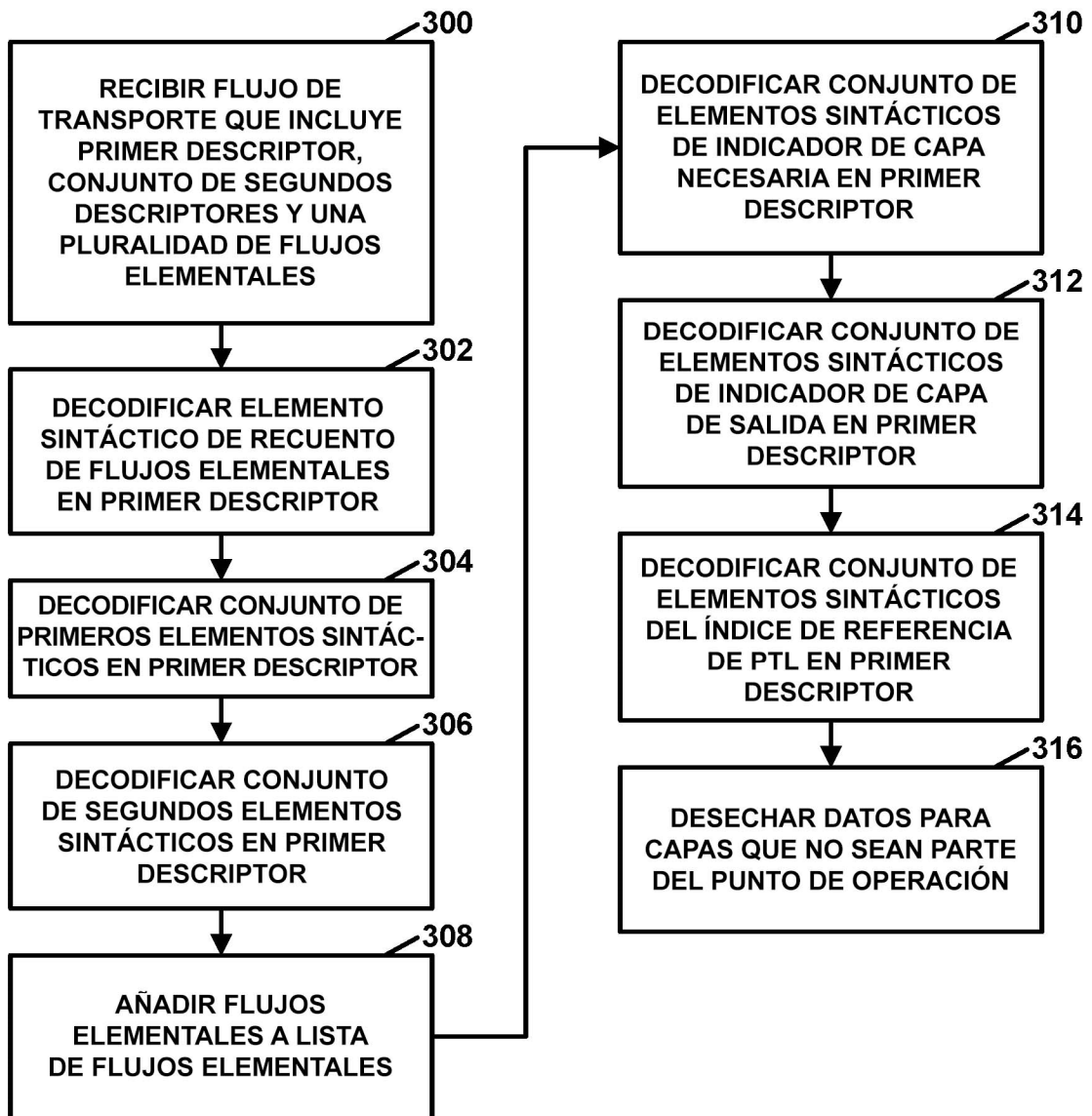


FIG. 7