

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 741 777**

51 Int. Cl.:

**H04N 19/597** (2014.01)

**H04N 19/70** (2014.01)

**H04N 19/30** (2014.01)

**H04N 19/187** (2014.01)

**H04N 19/169** (2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **09.10.2015 PCT/US2015/054865**

87 Fecha y número de publicación internacional: **14.04.2016 WO16057884**

96 Fecha de presentación y número de la solicitud europea: **09.10.2015 E 15787083 (3)**

97 Fecha y número de publicación de la concesión europea: **15.05.2019 EP 3205105**

54 Título: **Punto operativo para el transporte de flujos de bits en capas de la HEVC**

30 Prioridad:

**10.10.2014 US 201462062681 P**

**15.10.2014 US 201462064428 P**

**08.10.2015 US 201514878783**

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

**12.02.2020**

73 Titular/es:

**QUALCOMM INCORPORATED (100.0%)**

**International IP Administration,**

**5775 Morehouse Drive**

**San Diego, California 92121-1714, US**

72 Inventor/es:

**HENDRY, FNU;**

**CHEN, YING y**

**WANG, YE-KUI**

74 Agente/Representante:

**FORTEA LAGUNA, Juan José**

ES 2 741 777 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

Punto operativo para el transporte de flujos de bits en capas 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, incluidos 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, radioteléfonos celulares o por satélite, los llamados "teléfonos inteligentes", dispositivos de teleconferencia por vídeo, dispositivos de transmisión continua 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, UIT-T H.263, ITU-T H.264/MPEG-4, Parte 10, Codificación de vídeo avanzada (AVC) y UIT-T H.265, codificación de vídeo de alta eficacia (HEVC) y extensiones de dichas normas. Los dispositivos de vídeo pueden transmitir, recibir, codificar, decodificar y/o almacenar información de vídeo digital de manera más eficaz mediante la implementación de tales técnicas de codificación de vídeo.

**[0003]** Las técnicas de codificación de vídeo incluyen la predicción espacial (intraimagen) y/o la predicción temporal (interimagen) para reducir o eliminar la redundancia inherente en las secuencias de vídeo. Para la codificación de vídeo basada en bloques, un fragmento de vídeo (por ejemplo, un trama de vídeo o una parte de un 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 la predicción espacial con respecto a las muestras de referencia en bloques vecinos en la misma imagen. Los bloques de vídeo en un fragmento intercodificado (P o B) de una imagen pueden usar la predicción espacial con respecto a las muestras de referencia en bloques vecinos en la misma imagen o la predicción temporal con respecto a las 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 a 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 a una modalidad de intracodificación y a los datos residuales. Para una compresión adicional, los datos residuales pueden transformarse desde el dominio de píxeles a un dominio de transformación, dando como resultado coeficientes de transformación residuales, que luego pueden cuantizarse. Los coeficientes de transformación cuantizados, inicialmente dispuestos en una matriz bidimensional, pueden explorarse para producir un vector unidimensional de coeficientes de transformación, y puede aplicarse la codificación por entropía para lograr incluso más compresión.

**[0005]** El documento "Carriage of HEVC extension streams with MPEG-2 Systems [Transporte de flujos de extensión de la HEVC con sistemas MPEG-2]" de Chen Y et al describe una propuesta para el transporte de flujos de bits de extensión de la HEVC de múltiples capas, incluidos flujos de bits de SHVC y de MV-HEVC. Las propuestas incluyen un descriptor de HEVC, que puede incluirse como parte del descriptor de vídeo de HEVC. El descriptor de la HEVC señala los puntos operativos, cada uno de los cuales corresponde a una capa de salida establecida en la MV-HEVC/SHVC.

**[0006]** La solicitud de patente internacional con el número de publicación WO 2014/107312 A1 describe las actualizaciones de la estructura sintáctica nivel\_grada\_perfil para proporcionar más flexibilidad para el uso en la estructura sintáctica extensiones\_vps () para su uso en cada capa o punto operativo.

**[0007]** El documento "Proposed Study Text of ISO/IEC 13818-1:2013/PDAM7 [ Texto de estudio propuesto de ISO/IEC 13818-1: 2013/PDAM7]" por Karsten Grünberg et al describe las mejoras técnicas propuestas para el texto PDAM propuesto, incluso para la señalización de los puntos operativo.

**[0008]** El documento "Comments on Text of ISO/IEC 13818-1:2013 / PDAM 7 - Carriage of Layered HEVC [ Comentarios sobre el texto de ISO/IEC 13818-1: 2013/PDAM 7 - Transporte de la HEVC en capas]" por Madec Gérard describe los enfoques propuestos para la señalización de los puntos operativos, incluidos el descriptor\_extensión\_HEVC y el "descriptor\_punto\_operativo\_HEVC".

**SUMARIO**

**[0009]** El alcance de la protección está definido por las reivindicaciones independientes. Se incluyen características optativas en las reivindicaciones dependientes.

5 **[0010]** En general, esta divulgación describe técnicas para transportar datos de vídeo codificados de acuerdo, por ejemplo, a los Sistemas MPEG-2 (Grupo de Expertos en Imágenes en Movimiento). El transporte de datos de vídeo codificados también puede denominarse traslado de datos de vídeo codificados. Por ejemplo, la divulgación describe técnicas ejemplares, que pueden ser mejoras, del descriptor del flujo de transporte (TS) MPEG-2, que se utiliza para la señalización de la información de dependencia entre capas de flujos de bits de la HEVC (codificación de vídeo de alta eficacia) en capas. 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 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).

15 **[0011]** De acuerdo a un aspecto de la invención reivindicada, se proporciona un procedimiento, para procesar un flujo de bits que incluye datos de vídeo con codificación de vídeo de alta eficacia, HEVC, que incluye extraer un descriptor desde el flujo de bits, en donde el flujo de bits incluye capas de datos de vídeo para puntos operativos, por separado del descriptor, de modo que cada punto operativo incluya una o más de las capas de datos de vídeo, en donde la extracción del descriptor incluye: extraer un conjunto de estructuras de perfil, grada y nivel (PTL); y extraer datos que asocian cada una de las capas de cada uno de los puntos operativos con una correspondiente de las estructuras de PTL, en donde los datos que asocian cada una de las capas con la correspondiente de las estructuras de PTL son independientes de las estructuras de PTL en el descriptor; extraer datos de vídeo para uno de los puntos operativos del flujo de bits, basándose, al menos en parte, en las estructuras de PTL a las que corresponden las capas de uno de los puntos operativos; y proporcionar los datos de vídeo extraídos a un decodificador de vídeo.

25 **[0012]** De acuerdo a otro aspecto de la invención reivindicada, un dispositivo, para procesar un flujo de bits que incluye datos de vídeo con codificación de vídeo de alta eficacia, HEVC, incluye una memoria para almacenar datos extraídos desde el flujo de bits, y una o más unidades de procesamiento configuradas extraen un descriptor desde el flujo de bits, en donde el flujo de bits incluye capas de datos de vídeo para los puntos operativos, por separado del descriptor, de manera que cada punto operativo incluye una o más de las capas de datos de vídeo, en donde para extraer el descriptor, los uno o más procesadores están configurados para: extraer un conjunto de estructuras de perfil, grada y nivel (PTL), y extraer datos que asocian cada una de las capas de cada uno de los puntos operativos con una correspondiente de las estructuras de PTL, en donde los datos que asocian cada una de las capas con la correspondiente estructura de PTL son independientes de las estructuras de PTL en el descriptor, extraer datos de vídeo para uno de los puntos operativos del flujo de bits basándose, al menos en parte, en las estructuras de PTL a las que corresponden las capas de uno de los puntos operativos, y proporcionar los datos de vídeo extraídos a un decodificador de vídeo.

40 **[0013]** De acuerdo a otro aspecto de la invención reivindicada, un medio de almacenamiento legible por ordenador que tiene almacenadas en el mismo instrucciones que, cuando se ejecutan, hacen que un procesador extraiga un descriptor desde un flujo de bits que incluye datos de vídeo con codificación de vídeo de alta eficacia, HEVC, en donde el flujo de bits incluye capas de datos de vídeo para puntos operativos, por separado del descriptor, de modo que cada punto operativo incluya una o más capas de datos de vídeo, en donde las instrucciones que hacen que el procesador extraiga el descriptor comprenden instrucciones que hacen que el procesador: extraiga un conjunto de estructuras de perfil, grada y nivel (PTL); y extraiga los datos que asocian cada una de las capas de cada uno de los puntos operativos con una correspondiente de las estructuras de PTL, y en donde los datos que asocian cada una de las capas con la correspondiente de las estructuras de PTL son independientes de las estructuras de PTL en el descriptor; extraiga datos de vídeo para uno de los puntos operativos del flujo de bits basándose, al menos en parte, en las estructuras de PTL a las que corresponden las capas de uno de los puntos operativos; y proporcione los datos de vídeo extraídos a un decodificador de vídeo.

55 **[0014]** Los detalles de uno o más modos se exponen en los dibujos adjuntos y en la descripción siguiente. Otras características, objetos y ventajas de la invención serán evidentes a partir de la descripción y los dibujos y a partir de las reivindicaciones.

**BREVE DESCRIPCIÓN DE LOS DIBUJOS**

**[0015]**

60 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 a extensiones de una norma de codificación de vídeo.

65 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 a las extensiones de una 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 a extensiones de una norma de codificación de vídeo.

5 La figura 4 es un diagrama de bloques que ilustra un sistema ejemplar en el que un dispositivo de origen de audio/vídeo (A/V) transporta datos de audio y vídeo hasta un dispositivo de destino de A/V.

La figura 5 es un diagrama conceptual que ilustra un ejemplo de dependencia entre flujos elementales.

10 La figura 6 es otro diagrama conceptual que ilustra un ejemplo de dependencia entre flujos elementales.

La figura 7 es otro diagrama conceptual que ilustra un ejemplo de dependencia entre flujos elementales.

15 La figura 8 es un diagrama conceptual que ilustra un ejemplo de múltiples capas de datos de vídeo.

La figura 9 es un diagrama de flujo que ilustra un procedimiento ejemplar de acuerdo a las técnicas de esta divulgación.

## 20 DESCRIPCIÓN DETALLADA

[0016] En general, esta divulgación describe técnicas relacionadas con datos de nivel de los Sistemas del Grupo de Expertos en Imágenes en Movimiento (MPEG)-2 para datos de medios. Los sistemas MPEG-2 generalmente describen 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 sistemas MPEG-2 para datos de vídeo de múltiples capas. Más en particular, esta divulgación describe un descriptor del flujo de transporte (TS) de MPEG-2 que se puede usar para señalar información de dependencia entre capas de flujos de bits de HEVC en capas (por ejemplo, describir la dependencia (o relación) entre capas de flujos de bits de HEVC en capas, que puede ser mejoras relativas a algunas técnicas existentes). El documento de la norma HEVC se publica como ITU-T H.265, Serie H: Sistemas audiovisuales y de multimedios, Infraestructura de servicios audiovisuales - Codificación de vídeo en movimiento, Codificación de vídeo de alta eficacia, Sector de Normalización de la Telecomunicación de la Unión Internacional de Telecomunicación (UIT), abril de 2015.

[0017] Las técnicas de esta divulgación están generalmente dirigidas al traslado (por ejemplo, transporte) de datos de vídeo codificados de acuerdo a una extensión para una norma de codificación de vídeo (por ejemplo, una extensión para la norma de codificación de vídeo de alta eficacia (HEVC), la norma de HEVC también mencionada como UIT-T H.265). Dichas extensiones pueden incluir extensiones de múltiples vistas, tridimensionales y/o ajustables a escala. Por lo tanto, las técnicas de esta divulgación pueden aplicarse a la HEVC de múltiples vistas (MV-HEVC), la HEVC tridimensional (3D-HEVC) y la HEVC ajustable a escala (SHVC).

[0018] Los datos de vídeo de múltiples capas, por ejemplo, datos de vídeo de múltiples vistas y/o datos de vídeo con múltiples capas ajustables a escala, pueden incluir puntos operativos designados. En general, un punto operativo incluye un subconjunto de capas (por ejemplo, vistas) de un conjunto completo de capas de datos de vídeo de múltiples capas. El punto operativo 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 operativo solo para su uso como capa de referencia y, por lo tanto, dicha capa no se consideraría una capa de salida de destino.

[0019] Un tipo de dimensión ajustable a escala es la dimensión temporal. 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 del 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 del nivel temporal 2, para dar soporte a 60 FPS, y así sucesivamente. Un identificador temporal, o IDTemporal, puede ser señalado como representativo del nivel temporal al que pertenece una imagen en particular.

[0020] Un dispositivo de destino puede usar los descriptores de punto operativo incluidos en un flujo de bits para seleccionar uno de los puntos operativos a decodificar y finalmente presentar (por ejemplo, exhibir) a un usuario. En lugar de pasar datos para todas las vistas a un decodificador de vídeo una vez recibidos, el dispositivo de destino puede enviar solo las vistas de un punto operativo seleccionado al decodificador de vídeo. De esta manera, el dispositivo de destino puede descartar datos para vistas que no se decodificarán. Adicional o alternativamente, un dispositivo de red intermedio puede descartar datos para vistas que no corresponden a un punto operativo solicitado, por ejemplo, para utilizar mejor el ancho de banda. El dispositivo de destino puede seleccionar un punto operativo basándose en la calidad más alta con soporte en uno de los puntos operativos para un flujo de bits y/o basándose en una cantidad disponible de ancho de banda de red.

- 5 **[0021]** Los datos de vídeo también se pueden describir por perfiles, capas y gradas. Un "perfil" es un subconjunto de una sintaxis completa del flujo de bits que está especificado por una norma aplicable de codificación de vídeo. Un "nivel" corresponde a las limitaciones del consumo de recursos del decodificador, tales como, por ejemplo, la memoria y el cálculo del decodificador, que están relacionados con la resolución de las imágenes, la velocidad de bits y la velocidad de procesamiento de bloques.
- 10 **[0022]** 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, UIT-T H.263, ISO/IEC MPEG-4 Visual e ITU -T H.264 (también conocida como ISO/IEC MPEG-4 AVC), incluidas sus extensiones de codificación de vídeo ajustable a escala (SVC) y de codificación de vídeo de vista múltiple (MVC).
- 15 **[0023]** Recientemente, el Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) del Grupo de Expertos en Codificación de vídeo (VCEG) del UIT-T y el Grupo de Expertos en Imágenes en Movimiento (MPEG) de ISO/IEC ha finalizado el diseño de una nueva norma de codificación de vídeo, denominada Codificación de vídeo de Alta Eficacia (HEVC), la extensión de múltiples vistas de la HEVC, llamada MV-HEVC, y la extensión ajustable a escala de la HEVC, llamada SHVC.
- 20 **[0024]** El más reciente borrador de especificación de la HEVC, titulado "Draft high efficiency video coding (HEVC) version 2, combined format range extensions (RExt), scalability (SHVC), and multi-view (MV-HEVC) extensions [Borrador de codificación de vídeo de alta eficacia (HEVC), versión 2, extensiones combinadas de rango de formato (RExt), ajustabilidad a escala (SHVC) y vista múltiple (MV-HEVC)]" para el JCT-VC de UIT-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 18ª Conferencia: Sapporo, JP, del 30 de junio al 9 de julio de 2014 (JCTVC-R1013\_v6), está disponible en [phenix.intevry.fr/jct/doc\\_end\\_user/documents/18\\_Sapporo/wg11/JCTVC-R1013-v6.zip](http://phenix.intevry.fr/jct/doc_end_user/documents/18_Sapporo/wg11/JCTVC-R1013-v6.zip).
- 25 **[0025]** El más reciente borrador de especificación de la MV-HEVC, titulado "MV-HECV Draft Text 9 [Borrador de texto 9 de la MV-HECV]" para el Equipo de Colaboración Conjunta sobre las Extensiones de Codificación de vídeo tridimensional del UIT-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11 9ª Conferencia: Sapporo, JP, 3 a 9 de julio de 2014 (JCT3V-I1002-v7), está disponible en [phenix.intevry.fr/jct3v/doc\\_end\\_user/documents/9\\_Sapporo/wg11/JCT3V-I1002-v7.zip](http://phenix.intevry.fr/jct3v/doc_end_user/documents/9_Sapporo/wg11/JCT3V-I1002-v7.zip).
- 30 **[0026]** El borrador de especificación de la SHVC, titulado "High efficiency video coding (HEVC) scalable extension Draft 7 [Extensión ajustable a escala de codificación de vídeo de alta eficacia (HEVC), borrador 7]" para el JCT-VC de ITU-T SG 16 WP 3 e ISO/CEI JTC 1/SC 29/WG 11 18ª Conferencia: Sapporo, JP, 30 de junio a 9 de julio de 2014 (JCTVC-R1008v7) está disponible en [phenix.intevry.fr/jct/doc\\_end\\_user/documents/18\\_Sapporo/wg11/JCTVC-R1008-v7.zip](http://phenix.intevry.fr/jct/doc_end_user/documents/18_Sapporo/wg11/JCTVC-R1008-v7.zip).
- 35 **[0027]** Las tecnologías de Sistemas del MPEG-2 (Grupo de Expertos en Imágenes en Movimiento) se pueden emplear para transportar datos de vídeo. Los sistemas del MPEG-2 a veces se denominan MPEG-2 TS. La más reciente especificación de MPEG-2 TS es la recomendación UIT-T H.222.0, versión de junio de 2012, disponible en [www.itu.int/rec/T-REC-H.222.0-201206-l/en](http://www.itu.int/rec/T-REC-H.222.0-201206-l/en), con cargo. Esta especificación proporciona soporte para la AVC y las extensiones de la AVC.
- 40 **[0028]** La enmienda de MPEG-2 TS para la HEVC ha sido desarrollada. El más reciente documento es "Text of ISO/IEC 13818-1: 2013 / Final Draft Amendment 3 - Transport of HEVC video over MPEG-2 Systems [Texto de ISO/IEC 13818-1: 2013/Borrador final de la Enmienda 3 - Transporte de vídeo HEVC sobre sistemas MPEG-2]", en el documento de MPEG w13656, julio de 2013. Recientemente, se ha iniciado la enmienda de MPEG-2 TS para el traslado de la HEVC en capas. El más reciente documento es "Text of ISO/IEC 13818-1:2013 / Study of PDAM 7 - Carriage of Layered HEVC [Texto de ISO/IEC 13818-1: 2013 / Estudio de PDAM 7 - Traslado de HEVC en capas]", en el documento MPEG w14562, julio de 2014, y se denominará borrador L-HEVC TS en el presente documento. Un ejemplo del diseño de MPEG-2 TS para el traslado de extensiones de la HEVC se describe en la Solicitud de Patente Provisional Estadounidense N° 61/925.191, presentada el 8 de enero de 2014. Otro ejemplo del diseño de MPEG-2 TS para el traslado de extensiones de la HEVC se describe en la Solicitud de Patente Provisional Estadounidense N° 62/025.432, presentada el 16 de julio de 2014.
- 45 **[0029]** La especificación de sistemas MPEG-2 describe cómo los flujos de datos de multimedia comprimidos (vídeo y/o audio) se pueden multiplexar junto con otros datos para formar un solo flujo de datos adecuado para su transmisión o almacenamiento digital. Los sistemas MPEG-2 describen un flujo elemental, que es un componente único, codificado digitalmente (posiblemente comprimido según MPEG), de un programa. Por ejemplo, la parte codificada de vídeo o audio 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, un elemento sintáctico de identificador de flujo se utiliza para distinguir los paquetes de PES que pertenecen a un flujo elemental o a otro.
- 50 **[0030]** El flujo de programa y el flujo de transporte son dos multiplexados alternativos que se dirigen a diferentes aplicaciones. El flujo de programa está sesgado hacia el almacenamiento y la exhibición de un solo programa desde
- 55
- 60
- 65

un servicio de almacenamiento digital y un flujo de programa está concebido para su uso en entornos libres de errores, ya que puede ser susceptible a errores. 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 de PES que se obtienen de los flujos elementales contribuyentes se organizan en 'cajetillas'. Una cajetilla incluye un encabezado de cajetilla, un encabezado de sistema optativo y cualquier número de paquetes de PES tomados de cualquiera de los flujos elementales contribuyentes, en cualquier orden. El encabezado del sistema contiene un resumen de las características del flujo del programa, tales como: su máxima velocidad de datos; el número de flujos elementales contribuyentes de audio y vídeo; información adicional de temporización. Un decodificador puede usar la información contenida en un encabezado del sistema para determinar si el decodificador es capaz de decodificar el flujo del programa o no.

**[0031]** El flujo de transporte está concebido para la entrega simultánea de una serie de programas por canales potencialmente propensos a errores. Es un multiplex ideado para aplicaciones de múltiples programas, tales como la difusión, de modo que un solo flujo de transporte pueda alojar muchos programas independientes.

**[0032]** 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 del programa. Además, a cada paquete de transporte de 188 octetos se le otorga fácilmente protección adicional contra errores, procesándolo a través de un proceso de protección estándar contra errores, tal como la codificación de Reed-Solomon. La mejora de la capacidad de recuperación ante errores del flujo de transporte significa que tiene una mejor posibilidad de sobrevivir a los canales propensos a errores que se encuentran en un entorno de difusión, por ejemplo.

**[0033]** Podría parecer que el flujo de transporte es claramente el mejor de los dos multiplexados, con su mayor capacidad de recuperación a errores y su capacidad para transportar muchos programas simultáneos. Sin embargo, el flujo de transporte es un multiplex más sofisticado que el flujo de programa y, por consiguiente, es más difícil de crear y de demultiplexar.

**[0034]** El primer octeto de un paquete de transporte es un octeto de sincronización, que es 0x47 (es decir, el valor hexadecimal 47 o 0100 0111). Un solo flujo de transporte puede llevar muchos programas diferentes, comprendiendo cada uno de los cuales muchos flujos elementales empaquetados. Se utiliza un campo de identificador de paquete (PID) de 13 bits para distinguir los paquetes de transporte que contienen los datos de un flujo elemental de los que llevan los datos de otros flujos elementales. Es responsabilidad del multiplexor garantizar que a cada flujo elemental se le conceda un valor de PID único. El último octeto de un paquete de transporte es un campo de recuento de continuidad. Se incrementa entre los paquetes de transporte sucesivos que pertenecen al mismo flujo elemental. Esto permite que un decodificador detecte la pérdida o ganancia de un paquete de transporte y, con suerte, oculte los errores que de lo contrario podrían resultar de tal suceso.

**[0035]** Aunque el valor de PID deja claro a qué flujo elemental pertenece un paquete de transporte, el decodificador también puede necesitar determinar qué flujos elementales pertenecen a qué programa. La información específica del programa se utiliza para especificar explícitamente la relación entre los programas y los flujos elementales componentes. La información específica del programa puede incluir una tabla de mapas del programa (PMT), un mapa de flujos de programa (PSM), una tabla de asociaciones de programa (PAT), una tabla de información de red (NIT) y/o una tabla de acceso condicional (CAT).

**[0036]** Cada programa transportado en un flujo de transporte tiene asociada una Tabla de Mapas de Programa. Esta tabla proporciona detalles sobre el programa y los flujos elementales que forman el programa. Por ejemplo, puede haber un programa con el número 3 que contenga vídeo con PID 33, audio en inglés con PID 57 y audio en chino con PID 60. Se permite que una PMT incluya más de un programa. La tabla básica de mapas de programa puede ser embellecida con algunos de los muchos descriptores especificados dentro de la especificación de los sistemas MPEG-2. Los descriptores transmiten información adicional sobre un programa o sus flujos elementales componentes. Los descriptores pueden incluir, por ejemplo, parámetros de codificación de vídeo, parámetros de codificación de audio, identificación de idioma, información de exploración y recorrido, detalles de acceso condicional, información de derechos de autor, etc. Un emisor u otro usuario puede definir descriptores privados adicionales si es necesario. En los flujos elementales componentes relacionados con el vídeo, también hay un descriptor de jerarquía, que proporciona información para identificar los elementos de programa que contienen componentes de vídeo, audio y flujos privados codificados jerárquicamente.

**[0037]** El PSM proporciona una descripción de los flujos elementales en el Flujo de programas y su relación mutua. Cuando se transporta en un Flujo de Transporte, esta estructura no deberá modificarse, según la especificación de Sistemas MPEG-2. El PSM está presente como un paquete de PES cuando el valor de id\_flujo es 0xBC (valor hexadecimal BC, o 1011 1100).

**[0038]** En la Tabla de asociación de programas se mantiene una lista completa de todos los programas disponibles en un flujo de transporte. Esta tabla se puede encontrar fácilmente, ya que siempre tiene el valor de PID 0. Cada programa se enumera junto con el valor de PID de los paquetes de transporte que contienen su Tabla de mapas de

programa. Utilizando el mismo ejemplo mencionado anteriormente, la PMT que especifica los flujos elementales del programa número 3 tiene un PID de 1001 y otra PMT tiene otro PID de 1002. Este conjunto de información se incluye en la PAT.

5 **[0039]** El programa número cero, especificado en la PAT, tiene un significado especial. Este programa se utiliza para señalar el camino a la Tabla de información de red. La NIT es opcional. Cuando está presente, la NIT está destinada a proporcionar información sobre la red física que lleva el flujo de transporte, tal como las frecuencias de canal, los detalles del transpondedor del satélite, las características de modulación, el originador del servicio, el nombre del servicio y los detalles de las redes alternativas disponibles.

10 **[0040]** Si se aleatoriza cualquier flujo elemental dentro de un flujo de transporte, entonces debe estar presente una Tabla de acceso condicional, según la especificación de los Sistemas MPEG-2. La CAT proporciona detalles de los uno o más sistemas de aleatorización en uso y proporciona los valores de PID de los paquetes de transporte que contienen la información de administración y prestaciones de acceso condicional. El formato de esta información no se especifica en la especificación de Sistemas MPEG-2.

15 **[0041]** En MPEG-2 TS, un descriptor de jerarquía está diseñado para señalar la jerarquía de subflujos de bits en diferentes flujos elementales. El descriptor de jerarquía proporciona información para identificar los elementos de programa que contienen componentes de flujos de vídeo, de audio y privados, codificados jerárquicamente. La tabla 2-49 de la especificación de Sistemas MPEG-2 se reproduce a continuación. Como se describe con más detalle, en algunos ejemplos, la divulgación describe un descriptor de jerarquía mejorado y una actualización de la Tabla 2-49 que se muestra inmediatamente a continuación. La Tabla 2-49 actualizada para el descriptor de jerarquía mejorado se describe con más detalle más adelante.

25

**Tabla 2-49 - Descriptor de jerarquía**

Sintaxis	Nº de bits	Mnemónico
descriptor_jerarquia () {		
<b>etiqueta_descriptor</b>	<b>8</b>	<b>uimsbf</b>
<b>longitud_descriptor</b>	<b>8</b>	<b>uimsbf</b>
<b>reservado</b>	<b>1</b>	<b>bslbf</b>
<b>indicador_ajustabilidad_temporal_escala</b>	<b>1</b>	<b>bslbf</b>
<b>indicador_ajustabilidad_espacial_escala</b>	<b>1</b>	<b>bslbf</b>
<b>indicador_ajustabilidad_calidad_escala</b>	<b>1</b>	<b>bslbf</b>
<b>tipo_jerarquia</b>	<b>4</b>	<b>uimsbf</b>
<b>reservado</b>	<b>2</b>	<b>bslbf</b>
<b>índice_capa_jerarquia</b>	<b>6</b>	<b>uimsbf</b>
<b>indicador_tref_presente</b>	<b>1</b>	<b>bslbf</b>
<b>reservado</b>	<b>1</b>	<b>bslbf</b>
<b>índice_capa_integrada_jerarquia</b>	<b>6</b>	<b>uimsbf</b>
<b>reservado</b>	<b>2</b>	<b>bslbf</b>
<b>canal_jerarquia</b>	<b>6</b>	<b>uimsbf</b>
}		

**[0042]** La semántica para los elementos sintácticos de la Tabla 2-49 de los Sistemas MPEG-2 se proporciona a continuación:

30 **indicador\_ajustabilidad\_temporal\_escala:** un indicador de 1 bit que, cuando se fija en '0', indica que el elemento de programa asociado mejora la velocidad de tramas del flujo de bits resultante del elemento de programa al que hace referencia el índice\_capa\_integrada\_jerarquia. El valor de '1' para este indicador está reservado.

**indicador\_ajustabilidad\_espacial\_escala:** un indicador de 1 bit que, cuando se fija en '0', indica que el elemento de programa asociado mejora la resolución espacial del flujo de bits resultante del elemento de programa al que hace referencia el índice\_capa\_integrada\_jerarquia. El valor de '1' para este indicador está reservado.

35 **indicador\_ajustabilidad\_calidad\_escala:** un indicador de 1 bit que, cuando se fija en '0', indica que el elemento de programa asociado mejora la calidad o fidelidad de la SNR del flujo de bits resultante del elemento de programa al que hace referencia el índice\_capa\_integrada\_jerarquia. El valor de '1' para este indicador está reservado.

**tipo\_jerarquia:** la relación jerárquica entre la capa jerárquica asociada y su capa integrada de jerarquía se define

en la Tabla 2-50. Si la ajustabilidad a escala se aplica en más de una dimensión, este campo se fijará en el valor de '8' ("Ajustabilidad a escala combinada"), y los indicadores indicador\_ajustabilidad\_temporal\_escala, indicador\_ajustabilidad\_espacial\_escala e indicador\_ajustabilidad\_calidad\_escala se fijarán en consecuencia. Para los subflujos de bits de vídeo de la MVC, este campo se fijará en el valor de '9' ("Subflujo de bits de vídeo de MVC") y los indicadores indicador\_ajustabilidad\_temporal\_escala, indicador\_ajustabilidad\_espacial\_escala e indicador\_ajustabilidad\_calidad\_escala se fijarán en '1'. Para los subflujos de bits de vista base de la MVC, este campo se fijará en el valor de '15' y los indicadores indicador\_ajustabilidad\_temporal\_escala, indicador\_ajustabilidad\_espacial\_escala e indicador\_ajustabilidad\_calidad\_escala se fijarán en '1'.

**índice\_capa\_jerarquía:** el índice\_capa\_jerarquía 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 subflujos de bits de vídeo de flujos de vídeo de la AVC que son conformes a uno o más perfiles definidos en el Anexo G de la Rec. UIT-T H.264 | ISO/CEI 14496-10, este es el índice de elemento de programa, que se asigna de manera que el orden del flujo de bits sea correcto si las representaciones de dependencia de SVC asociadas de los subflujos de bits de vídeo de la misma unidad de acceso se vuelven a ensamblar en orden creciente del índice\_capa\_jerarquía. Para los subflujos de bits de vídeo de la MVC de flujos de vídeo de la AVC que son conformes a uno o más perfiles definidos en el Anexo H de la Rec. UIT-T H.264 | ISO/CEI 14496-10, este es el índice de elemento de programa, que se asigna de manera que el orden del flujo de bits será correcto si los subconjuntos de componentes de vista de MVC asociados de los subflujos de bits de vídeo de MVC de la misma unidad de acceso se vuelven a ensamblar en orden creciente del índice\_capa\_jerarquía.

**indicador\_tref\_presente:** un indicador de 1 bit que, cuando se fija en '0', indica que el campo TREF puede estar presente en los encabezados de paquetes de PES en el flujo elemental asociado. El valor de '1' para este indicador está reservado.

**índice\_capa\_integrada\_jerarquía - índice\_capa\_integrada\_jerarquía** es un campo de 6 bits que define el índice\_capa\_jerarquía del elemento de programa al que se debe acceder y que debe estar presente en el orden de decodificación antes de decodificar el flujo elemental asociado a este descriptor\_jerarquía. Este campo está indefinido si el valor de tipo\_jerarquía es 15.

**canal\_jerarquía:** canal\_jerarquía es un campo de 6 bits que indica el número de canal previsto para el elemento de programa asociado en un conjunto ordenado de canales de transmisión. El canal de transmisión más robusto está definido por el valor más bajo de este campo con respecto a la definición general de la jerarquía de transmisión. Un determinado canal\_jerarquía puede ser asignado al mismo tiempo a varios elementos de programa.

La tabla 2-50 de la especificación de Sistemas MPEG-2 se reproduce inmediatamente a continuación. En algunos ejemplos, la divulgación describe las actualizaciones de la Tabla 2-50 como parte de la descripción de un descriptor de jerarquía mejorado. Se describe la Tabla 2-50 actualizada con más detalle a continuación más adelante.

**Tabla 2-50 - valores del campo tipo\_jerarquía**

Valor	Descripción
0	Reservado
1	Escalabilidad espacial
2	Escalabilidad de SNR
3	Escalabilidad temporal
4	División de datos
5	Flujo de bits de extensión
6	Flujo privado
7	Perfil de múltiples vistas
8	Escalabilidad combinada
9	subflujo de bits de vídeo de MVC
10 a 14	Reservado
15	Capa base o subflujo de bits de vista base de MVC o subflujo de bits de vídeo de AVC de la MVC

**[0043]** En MPEG-2 TS, dos descriptores están diseñados para señalar las características de los subflujos de bits para la SVC y la MVC, respectivamente: Descriptor de extensión de SVC y descriptor de extensión de MVC. La SVC y la MVC son las extensiones de la codificación de vídeo ajustables a escala y de la codificación de vídeo de vista múltiple de la norma ITU-T H.264/AVC. Además, en MPEG-2 TS, hay un descriptor de punto operativo de la MVC que describe las características de los puntos operativos. La sintaxis y la semántica de los tres descriptores se proporcionan a continuación.



[0044] La Tabla 2-96 a continuación ilustra los elementos sintácticos para el Descriptor de Extensión de SVC de los Sistemas MPEG-2. Para subflujos de bits de vídeo de flujos de vídeo de la AVC que son conformes a uno o más perfiles definidos en el Anexo G de la Rec. ITU T H.264 | ISO/IEC 14496-10, el descriptor de la extensión de la SVC de la Tabla 2-96 proporciona información sobre el flujo de vídeo de AVC que resulta de reensamblar (hasta) el subflujo de bits de vídeo asociado y brinda información sobre la ajustabilidad a escala y el reensamblaje del subflujo de bits de vídeo. Puede haber un descriptor de extensión de SVC, asociado a cualquiera de las secuencias de subflujos de bits de vídeo de un flujo de vídeo de AVC, que es conforme a uno o más perfiles definidos en el Anexo G de la Rec. UIT-T H.264 | ISO/IEC 14496-10.

Tabla 2-96 - descriptor de extensión de SVC

Sintaxis	Nº de bits	Mnemónico
Descriptor_extensión_SVC () {		
<b>etiqueta_descriptor</b>	<b>8</b>	<b>uimsbf</b>
<b>longitud_descriptor</b>	<b>8</b>	<b>uimsbf</b>
<b>anchura</b>	<b>16</b>	<b>uimsbf</b>
<b>altura</b>	<b>16</b>	<b>uimsbf</b>
<b>velocidad_tramas</b>	<b>16</b>	<b>uimsbf</b>
<b>velocidad_media_bits</b>	<b>16</b>	<b>uimsbf</b>
<b>velocidad_máxima_bits</b>	<b>16</b>	<b>uimsbf</b>
<b>id_dependencia</b>	<b>3</b>	<b>bslbf</b>
<b>reservado</b>	<b>5</b>	<b>bslbf</b>
<b>inicio_id_calidad</b>	<b>4</b>	<b>bslbf</b>
<b>fin_id_calidad</b>	<b>4</b>	<b>bslbf</b>
<b>inicio_id_temporal</b>	<b>3</b>	<b>bslbf</b>
<b>fin_id_temporal</b>	<b>3</b>	<b>bslbf</b>
<b>ninguna_unidad_nal_sei_presente</b>	<b>1</b>	<b>bslbf</b>
<b>reservado</b>	<b>1</b>	<b>bslbf</b>
}		

[0045] La semántica para los elementos sintácticos de la Tabla 2-96 de acuerdo a la especificación de Sistemas MPEG-2 se proporciona a continuación:

**anchura:** este campo de 16 bits indica la resolución máxima del ancho de imagen, en píxeles, del flujo reensamblado de vídeo de AVC.

**altura:** este campo de 16 bits indica la resolución máxima de altura de la imagen, en píxeles, del flujo reensamblado de vídeo de AVC.

**velocidad trama:** este campo de 16 bits indica la velocidad máxima de tramas, en tramas / 256 segundos, del flujo reensamblado de vídeo de AVC.

**velocidad media\_bits:** este campo de 16 bits indica la velocidad media de bits, en kbits por segundo, del flujo reensamblado de vídeo de AVC.

**velocidad máxima\_bits:** este campo de 16 bits indica la velocidad máxima de bits, en kbits por segundo, del flujo reensamblado de vídeo AVC.

**id\_dependencia:** este campo de 3 bits indica el valor del id\_dependencia asociado al subflujo de bits de vídeo.

**inicio\_id\_calidad:** este campo de 4 bits indica el valor mínimo del id\_calidad del elemento sintáctico del encabezado de la unidad de NAL de todas las unidades de NAL contenidas en el subflujo de bits de vídeo asociado.

**fin\_id\_calidad:** este campo de 4 bits indica el valor máximo del id\_calidad del elemento sintáctico de encabezado de unidad de NAL de todas las unidades de NAL contenidas en el subflujo de bits de vídeo asociado.

**inicio\_id\_temporal:** este campo de 3 bits indica el valor mínimo del id\_temporal del elemento sintáctico del encabezado de la unidad de NAL de todas las unidades de NAL contenidas en el subflujo de bits de vídeo asociado.

**fin\_id\_temporal:** este campo de 3 bits indica el valor máximo del id\_temporal del elemento sintáctico del encabezado de la unidad de NAL de todas las unidades de NAL contenidas en el subflujo de bits de vídeo asociado.

**ninguna\_unidad\_nal\_sei\_presente:** este indicador de 1 bit, cuando se fija en '1', indica que no hay unidades de NAL de SEI presentes en el subflujo asociado de bits de vídeo. En caso de que el indicador ninguna\_unidad\_nal\_sei\_presente esté fijado en '1' para todos los subflujos de bits de vídeo de SVC y no esté

fijado en '1' o no esté presente para el subflujo de bits de vídeo de AVC de la SVC, cualquier unidad de NAL de SEI, si está presente, se incluye en el subflujo de bits de vídeo de AVC de la SVC. Si el descriptor de la extensión de la SVC está ausente para todos los subflujos de bits de vídeo, las unidades de NAL de SEI pueden estar presentes en cualquier representación de dependencia de la SVC de un subflujo de bits de vídeo de SVC, y pueden requerir un reordenamiento según el orden de las unidades de NAL dentro de una unidad de acceso, según se define en Rec. UIT-T H.264 | ISO/IEC 14496-10 antes del reensamblaje de la unidad de acceso.

**[0046]** La tabla 2-97 a continuación proporciona la sintaxis para el descriptor de la extensión de la MVC de la especificación de los Sistemas MPEG-2. Para los subflujos de bits de vídeo de la MVC de flujos de vídeo de la AVC que son conformes a uno o más perfiles definidos en el Anexo H de la Rec. UIT-T H.264 | ISO/IEC 14496-10, el descriptor de la extensión de la MVC proporciona información sobre el flujo de vídeo de AVC que resulta del reensamblaje de (hasta) el subflujo de bits de vídeo MVC asociado y proporciona información sobre el subflujo contenido de bits de vídeo de la MVC y para el reensamblaje del subflujo de bits de vídeo de MVC asociado. Puede haber un descriptor de extensión de MVC asociado a cualquiera de los subflujos de bits de vídeo de la MVC (con tipo\_flujo igual a 0x20) de un flujo de vídeo de AVC que es conforme a uno o más perfiles definidos en el Anexo H de la Rec. UIT-T H.264 | ISO/IEC 14496 10. Cuando el subflujo de bits de vídeo de la MVC es un subflujo de bits de vista base de la MVC, el descriptor de extensión de la MVC estará presente en la PMT o el PSM asociados para tipo\_flujo igual a 0x1B.

**Tabla 2-97 - descriptor de extensión de MVC**

Sintaxis	Nº de bits	Mnemónico
descriptor_extensión_MVC () {		
<b>etiqueta_descriptor</b>	<b>8</b>	<b>uimsbf</b>
<b>longitud_descriptor</b>	<b>8</b>	<b>uimsbf</b>
<b>velocidad_media_bits</b>	<b>16</b>	<b>uimsbf</b>
<b>velocidad_máxima_bits</b>	<b>16</b>	<b>uimsbf</b>
<b>reservado</b>	<b>4</b>	<b>bslbf</b>
<b>mín_índice_orden_vista</b>	<b>10</b>	<b>bslbf</b>
<b>máx_índice_orden_vista</b>	<b>10</b>	<b>bslbf</b>
<b>inicio_id_temporal</b>	<b>3</b>	<b>bslbf</b>
<b>fin_id_temporal</b>	<b>3</b>	<b>bslbf</b>
<b>ninguna_unidad_nal_sei_presente</b>	<b>1</b>	<b>bslbf</b>
<b>ninguna_unidad_nal_prefijo_presente</b>	<b>1</b>	<b>bslbf</b>
}		

**[0047]** La semántica para los elementos sintácticos de la Tabla 2-97 de acuerdo a la especificación de Sistemas MPEG-2 se proporciona a continuación:

**velocidad\_media\_bits:** este campo de 16 bits indica la velocidad media de bits, en kbits por segundo, del flujo reensamblado de vídeo de la AVC. Cuando se fija en 0, no se indica la velocidad media de bits.

**velocidad\_máxima\_bits:** este campo de 16 bits indica la velocidad máxima de bits, en kbits por segundo, del flujo reensamblado de vídeo de la AVC. Cuando se fija en 0, no se indica la velocidad máxima de bits.

**mín\_índice\_orden\_vista:** este campo de 10 bits indica el valor mínimo del índice de orden de vista de todas las unidades de NAL contenidas en el subflujo asociado de bits de vídeo de la MVC.

**máx\_índice\_orden\_vista:** este campo de 10 bits indica el valor máximo del índice de orden de vista de todas las unidades de NAL contenidas en el subflujo asociado de bits de vídeo de la MVC.

**inicio\_id\_temporal:** este campo de 3 bits indica el valor mínimo del id\_temporal del elemento sintáctico del encabezado de la unidad de NAL de todas las unidades de NAL contenidas en el subflujo asociado de bits de vídeo de la MVC.

**fin\_id\_temporal:** este campo de 3 bits indica el valor máximo del identificador temporal del elemento sintáctico del encabezado de la unidad de NAL de todas las unidades de NAL contenidas en el subflujo asociado de bits de vídeo de la MVC.

**ninguna\_unidad\_nal\_sei\_presente:** este indicador de 1 bit, cuando se fija en '1', indica que no hay unidades de NAL de SEI presentes en el subflujo asociado de bits de vídeo. En caso de que el indicador **ninguna\_unidad\_nal\_sei\_presente** esté fijado en '1' para todos los subflujos de bits de vídeo de la MVC y no esté fijado en '1' o no esté presente para el subflujo de bits de vídeo de AVC de la MVC, cualquier unidad de NAL de SEI, si está presente, se incluye en el subflujo de bits de vídeo de AVC de la MVC. Si el descriptor de la extensión de la MVC está ausente para todos los subflujos de bits de vídeo de la MVC, las unidades de NAL de

SEI pueden estar presentes en cualquier subconjunto de componentes de vista de la MVC de un subflujo de bits de vídeo de MVC, y pueden requerir un reordenamiento según el orden de las unidades de NAL dentro de una unidad de acceso, como se define en la Rec. UIT-T H.264 | ISO/IEC 14496-10 antes del reensamblaje de la unidad de acceso.

5 **ninguna\_unidad\_nal\_prefijo\_presente:** este indicador de 1 bit, cuando se fija en '1', indica que no hay unidades de NAL con prefijo presentes en el subflujo de bits de vídeo de AVC de la MVC o los subflujos de bits de vídeo de la MVC. Cuando este bit se fija en '0', indica que las unidades de NAL con prefijo están presentes en el subflujo de bits de vídeo de AVC de la MVC solamente.

10 **[0048]** La tabla 2-100 a continuación proporciona la sintaxis para el descriptor del punto operativo de la MVC, de la especificación de Sistemas MPEG-2. El descriptor del punto operativo de la MVC (véase la Tabla 2-100) proporciona un procedimiento para indicar el perfil y el nivel de uno o más puntos operativos, cada uno de los cuales está constituido por un conjunto de uno o más subflujos de bits de vídeo de la MVC. Si está presente, el descriptor del punto operativo de la MVC se incluirá en el grupo de elementos de datos inmediatamente después del campo longitud\_información\_programa en la sección\_mapa\_programa. Si un descriptor de punto operativo de la MVC está presente dentro de una descripción de programa, al menos un descriptor de jerarquía estará presente para cada subflujo de bits de vídeo de MVC presente en el mismo programa. De acuerdo a la especificación de los Sistemas MPEG-2, para indicar diferentes perfiles, se utiliza un descriptor de punto operativo de MVC por cada perfil.

20

**Tabla 2-100 - Descriptor de punto operativo de MVC**

Sintaxis	Nº de bits	Mnemónico
descriptor_punto_operativo_MVC () {		
<b>etiqueta_descriptor</b>	<b>8</b>	<b>uimsbf</b>
<b>longitud_descriptor</b>	<b>8</b>	<b>uimsbf</b>
<b>idc_perfil</b>	<b>8</b>	<b>uimsbf</b>
<b>indicador_conjunto_restricciones0</b>	<b>1</b>	<b>bslbf</b>
<b>indicador_conjunto_restricciones1</b>	<b>1</b>	<b>bslbf</b>
<b>indicador_conjunto_restricciones2</b>	<b>1</b>	<b>bslbf</b>
<b>indicador_conjunto_restricciones3</b>	<b>1</b>	<b>bslbf</b>
<b>indicador_conjunto_restricciones4</b>	<b>1</b>	<b>bslbf</b>
<b>indicador_conjunto_restricciones5</b>	<b>1</b>	<b>bslbf</b>
<b>Indicadores_compatibilidad_AVC</b>	<b>2</b>	<b>bslbf</b>
<b>recuento_nivel</b>	<b>8</b>	<b>uimsbf</b>
para (recomendación = 0; recomendación <		
recuento_nivel; i ++) {	<b>8</b>	<b>uimsbf</b>
<b>idc_nivel</b>	<b>8</b>	<b>uimsbf</b>
<b>recuento_puntos_operativos</b>		
para (j = 0; j < recuento_puntos_operativos; j ++ ) {	<b>5</b>	<b>bslbf</b>
<b>reservado</b>	<b>3</b>	<b>uimsbf</b>
<b>id_temporal_aplicable</b>	<b>8</b>	<b>uimsbf</b>
<b>núm_vistas_salida_destino</b>	<b>8</b>	<b>uimsbf</b>
<b>recuento_ES</b>		
para (k = 0; k < recuento_ES; k ++ ) {	<b>2</b>	<b>bslbf</b>
<b>reservado</b>	<b>6</b>	<b>uimsbf</b>
<b>referencia_ES</b>		
}		
}		
}		
}		
}		

**[0049]** La semántica para los elementos sintácticos de la Tabla 2-100, de acuerdo a la especificación de Sistemas MPEG-2, se proporciona a continuación:

- 5 **idc\_perfil** - Este campo de 8 bits indica el perfil, como se define en la Rec. UIT-T H.264 | ISO/IEC 14496-10, de todos los puntos operativos descritos dentro de este descriptor para el flujo de bits de la MVC.
- indicador\_conjunto\_restricciones0, indicador\_conjunto\_restricciones1,**  
**indicador\_conjunto\_restricciones2, indicador\_conjunto\_restricciones3,**  
**indicador\_conjunto\_restricciones4, indicador\_conjunto\_restricciones5** - Estos campos se codificarán de acuerdo a la semántica para estos campos definidos en la Rec. UIT-T H.264 | ISO/IEC 14496-10.
- 10 **indicadores\_compatibilidad\_AVC** - La semántica de los indicadores\_compatibilidad\_AVC es exactamente igual a la semántica de los uno o más campos definidos para los 2 bits entre el indicador\_conjunto\_restricciones2 y el campo idc\_nivel en el conjunto de parámetros de secuencia, como se define en la Rec. UIT-T H.264 | ISO/IEC 14496-10.
- 15 **recuento\_nivel**: este campo de 8 bits indica la cantidad de niveles para los cuales se describen puntos operativos.
- idc\_nivel** - Este campo de 8 bits indica el nivel, como se define en la Rec. UIT-T H.264 | ISO/IEC 14496-10, del flujo de bits de la MVC para los puntos operativos descritos por los siguientes grupos de elementos de datos.
- recuento\_puntos\_operativos**: este campo de 8 bits indica el número de puntos operativos descritos por la lista incluida en el siguiente grupo de elementos de datos.
- 20 **id\_temporal\_aplicable**: este campo de 3 bits indica el valor más alto del identificador temporal de las unidades de NAL en el flujo reensamblado de vídeo de la AVC.
- núm\_vistas\_salida\_destino**: este campo de 8 bits indica el valor del número de vistas destinadas a la salida para el punto operativo asociado.
- 25 **recuento\_ES**: este campo de 8 bits indica el número de valores de referencia\_ES incluidos en el siguiente grupo de elementos de datos. Los flujos elementales indicados en el siguiente grupo de elementos de datos forman juntos un punto operativo del flujo de bits de vídeo de la MVC. El valor 0xff está reservado.
- referencia\_ES**: este campo de 6 bits indica el valor de índice de capa de jerarquía presente en el descriptor de jerarquía que identifica un subflujo de bits de vídeo. El perfil y el nivel para un solo punto operativo, por ejemplo, todo el flujo de bits de vídeo de la MVC, se puede señalar usando el descriptor de vídeo de la AVC. Más allá de eso, la MVC permite decodificar diferentes subconjuntos de vista que pueden requerir diferentes perfiles y/o niveles. La especificación del descriptor de punto operativo de la MVC presta soporte a la indicación de diferentes perfiles y niveles para múltiples puntos operativos.
- 30

35 **[0050]** La tabla Amd7-1 a continuación proporciona la sintaxis para el descriptor de vídeo de la HEVC, de acuerdo a la especificación de los Sistemas MPEG-2. Para un flujo de vídeo de la HEVC, el descriptor de vídeo de la HEVC proporciona información básica para identificar los parámetros de codificación, tales como los parámetros de perfil y nivel, de ese flujo de vídeo de la HEVC. Para un subflujo de bits de vídeo temporal de la HEVC o un subconjunto de vídeo temporal de la HEVC, el descriptor de vídeo de la HEVC proporciona información tal como la representación asociada más alta de subcapa temporal de la HEVC, contenida en el flujo elemental al que se aplica.

40 **[0051]** En el borrador L-HEVC TS, la información de perfil, grada y nivel (PTL) y la información de punto operativo se señalizan en un descriptor de extensión de la HEVC y en un descriptor de punto operativo de la HEVC.

**Tabla Amd7-1 - descriptor de vídeo de la HEVC**

Sintaxis	Nº de bits	Mnemónico
descriptor_HEVC () {		
<b>etiqueta_descriptor</b>	<b>8</b>	<b>uimsbf</b>
<b>longitud_descriptor</b>	<b>8</b>	<b>uimsbf</b>
<b>espacio_perfil</b>	<b>2</b>	<b>uimsbf</b>
<b>indicador_grada</b>	<b>1</b>	<b>bslbf</b>
<b>idc_perfil</b>	<b>5</b>	<b>uimsbf</b>
<b>indicación_compatibilidad_perfil</b>	<b>32</b>	<b>bslbf</b>
<b>indicador_origen_progresivo</b>	<b>1</b>	<b>bslbf</b>
<b>indicador_origen_entrelazado</b>	<b>1</b>	<b>bslbf</b>
<b>indicador_restricción_no_empaquetada</b>	<b>1</b>	<b>bslbf</b>
<b>indicador_restricción_trama_solamente</b>	<b>1</b>	<b>bslbf</b>
<b>reservado_cero_44bits</b>	<b>44</b>	<b>bslbf</b>
<b>idc_nivel</b>	<b>8</b>	<b>uimsbf</b>

Sintaxis	Nº de bits	Mnemónico
<b>indicador_subconjunto_capa_temporal</b>	<b>1</b>	<b>bslbf</b>
<b>Indicador_fija_HEVC_presente</b>	<b>1</b>	<b>bslbf</b>
<b>indicador_imagen_presente_24hr_HEVC</b>	<b>1</b>	<b>bslbf</b>
<b>reservado</b>	<b>5</b>	<b>bslbf</b>
si (indicador_subconjunto_capa_temporal == '1') {		
<b>reservado</b>	<b>5</b>	<b>bslbf</b>
<b>mín_id_temporal</b>	<b>3</b>	<b>uimsbf</b>
<b>reservado</b>	<b>5</b>	<b>bslbf</b>
<b>máx_id_temporal</b>	<b>3</b>	<b>uimsbf</b>
}		
}		

**[0052]** La semántica para los elementos sintácticos de la Tabla X-1, de acuerdo a la especificación de los Sistemas MPEG-2, se proporciona a continuación:

5 **espacio\_perfil, indicador\_grada, idc\_perfil, indicación\_compatibilidad\_perfil, indicador\_origen\_progresivo, indicador\_origen\_entrelazado, indicador\_restricción\_no\_empaquetada, indicador\_restricción\_trama\_solamente, reservado\_cero\_44bits, idc\_nivel** - Cuando el descriptor de vídeo de la HEVC se aplica a un flujo de vídeo de la HEVC o a una representación temporal completa de la HEVC, estos campos se codificarán de acuerdo a la semántica definida en la Rec. UIT-T H.265 | ISO/IEC 23008-2 para  
10 espacio\_perfil\_general, indicador\_grada\_general, idc\_perfil\_general, indicador\_compatibilidad\_perfil\_general[i], indicador\_origen\_progresivo\_general, indicador\_origen\_entrelazado\_general, indicador\_restricción\_empaquetada\_no\_general, indicador\_restricción\_general\_trama\_solamente, reservado\_general\_cero\_44bits, idc\_nivel\_general, respectivamente, para el correspondiente flujo de vídeo de la HEVC o la representación temporal completa de la HEVC, y el flujo completo de vídeo de la HEVC, o la representación temporal completa de la HEVC a la cual está asociado el descriptor de vídeo de la HEVC, serán conformes a la  
15 información señalizada por estos campos.

**[0053]** Cuando el descriptor de vídeo de la HEVC se aplica a un subflujo de bits de vídeo temporal de la HEVC o a un subconjunto de vídeo temporal de la HEVC, del cual la correspondiente representación de subcapa temporal más alta de la HEVC no es una representación temporal completa de la HEVC, estos campos se codificarán de acuerdo a la semántica definida en la Rec. UIT-T H.265 | ISO/IEC 23008-2 para *indicador\_perfil\_sub\_capa, indicador\_grada\_sub\_capa, idc\_perfil\_sub\_capa, indicador\_compatibilidad\_perfil\_sub\_capa [i], indicador\_origen\_progresivo\_sub\_capa, indicador\_origen\_entrelazado\_sub\_capa, indicador\_restricción\_no\_empaquetada\_sub\_capa, indicador\_restricción\_trama\_solamente\_sub\_capa, sub\_capa\_reservado\_cero\_44bits, idc\_nivel\_sub\_capa*, respectivamente, para la correspondiente representación más alta de subcapa temporal de la HEVC, y toda la más alta representación temporal de subcapa de la HEVC, a la que está asociado el descriptor de vídeo de la HEVC, será conforme a la información señalizada por estos campos.  
20  
25

**[0054]** En una o más secuencias en el flujo de vídeo de la HEVC, el nivel puede ser más bajo que el nivel señalado en el descriptor de vídeo de la HEVC, mientras que también puede ocurrir un perfil que es un subconjunto del perfil señalado en el descriptor de vídeo de la HEVC. Sin embargo, en todo el flujo de vídeo de la HEVC, solo se usarán los subconjuntos de toda la sintaxis del flujo de bits que se incluyen en el perfil señalado en el descriptor de vídeo de la HEVC, si está presente. Si los conjuntos de parámetros de secuencia en un flujo de vídeo de la HEVC señalizan diferentes perfiles, y no se señalizan restricciones adicionales, entonces el flujo puede necesitar un examen para determinar a qué perfil, si acaso, es conforme el flujo completo. Si un descriptor de vídeo de la HEVC ha de asociarse a un flujo de vídeo de la HEVC que no es conforme a un solo perfil, entonces el flujo de vídeo de la HEVC debería dividirse en dos o más subflujos, de modo que los descriptores de vídeo de la HEVC puedan señalar un solo perfil para cada subflujo de ese tipo.  
30  
35

**[0055] indicador\_subconjunto\_capa\_temporal:** este indicador de 1 bit, cuando se fija en '1', indica que los elementos sintácticos que describen un subconjunto de capas temporales se incluyen en este descriptor. Este campo se fijará en 1 para los subconjuntos de vídeo temporal de la HEVC y para los subflujos de bits de vídeo temporal de la HEVC. Cuando se fija en '0', los elementos sintácticos mín\_id\_temporal y máx\_id\_temporal no se incluyen en este descriptor.  
40  
45

**[0056] indicador\_fija\_HEVC\_presente:** este campo de 1 bit, cuando se fija en '1', indica que el flujo de vídeo de la HEVC o la representación de la subcapa temporal más alta de la HEVC pueden incluir imágenes fijas de la HEVC.

Cuando se fija en '0', entonces el flujo asociado de vídeo de la HEVC no contendrá imágenes fijas de la HEVC. Según la Rec. UIT-T H.265 | ISO/CEI 23008-2, las imágenes IDR siempre se asocian a un valor de IdTemporal igual a 0. En consecuencia, si el descriptor de vídeo de la HEVC se aplica a un subconjunto de vídeo temporal de la HEVC, las imágenes fijas de la HEVC solo pueden estar presentes en el subflujo temporal asociado de bits de vídeo de la HEVC.

5 **[0057] Indicador\_imagen\_presente\_24hr\_HEVC:** este indicador de 1 bit, cuando se fija en '1', indica que el flujo asociado de vídeo de la HEVC o la representación temporal más alta de la HEVC puede contener imágenes de 24 horas de la HEVC. Para la definición de una imagen de 24 horas de la HEVC, véase 2.1.97. Si este indicador se fija en '0', el flujo asociado de vídeo de la HEVC no contendrá ninguna imagen de 24 horas de la HEVC.

10 **[0058] mín\_id\_temporal:** este campo de 3 bits indica el valor mínimo de TemporalId, como se define en la Rec. UIT-T H.265 | ISO/IEC 23008-2, de todas las unidades de acceso de la HEVC en el flujo elemental asociado.

15 **[0059] máx\_id\_temporal -** Este campo de 3 bits indica el valor máximo de IdTemporal, como se define en la Rec. UIT-T H.265 | ISO/IEC 23008-2, de todas las unidades de acceso de la HEVC en el flujo elemental asociado.

**Tabla Amd7-2 - descriptor de punto operativo de HEVC**

Sintaxis	Nº de bits	Mnemónico
descriptor_punto_operativo_HEVC () {		
<b>etiqueta_descriptor</b>	<b>8</b>	<b>uimsbf</b>
<b>longitud_descriptor</b>	<b>8</b>	<b>uimsbf</b>
<b>espacio_perfil</b>	<b>2</b>	<b>uimsbf</b>
<b>indicador_grada</b>	<b>1</b>	<b>bslbf</b>
<b>idc_perfil</b>	<b>5</b>	<b>uimsbf</b>
<b>indicación_compatibilidad_perfil</b>	<b>32</b>	<b>bslbf</b>
<b>indicador_origen_progresivo</b>	<b>1</b>	<b>bslbf</b>
<b>indicador_origen_entrelazado</b>	<b>1</b>	<b>bslbf</b>
<b>indicador_restricción_no_empaquetada</b>	<b>1</b>	<b>bslbf</b>
<b>indicador_restricción_trama_solamente</b>	<b>1</b>	<b>bslbf</b>
<b>reservado_cero_44bits</b>	<b>44</b>	<b>bslbf</b>
<b>recuento_nivel</b>	<b>8</b>	<b>uimsbf</b>
para (i = 0; i < nivel_recuento; i++) {		
<b>idc_nivel</b>	<b>8</b>	<b>uimsbf</b>
<b>recuento_puntos_operativos</b>	<b>8</b>	<b>uimsbf</b>
para (j = 0; j < recuento_puntos_operativos; j++) {		
<b>reservado</b>	<b>3</b>	<b>bslbf</b>
<b>idc_información_velocidad_trama_constante</b>	<b>2</b>	<b>uimsbf</b>
<b>id_temporal_aplicable</b>	<b>3</b>	<b>uimsbf</b>
<b>núm_vistas_salida_destino</b>	<b>8</b>	<b>uimsbf</b>
<b>recuento_ES</b>	<b>8</b>	<b>uimsbf</b>
para (k = 0; k < recuento_ES; k++) {		
<b>reservado</b>	<b>1</b>	<b>bslbf</b>
<b>dependencias_antepuestas</b>	<b>1</b>	<b>bslbf</b>
<b>referencia_ES</b>	<b>6</b>	<b>uimsbf</b>
}		
}		
si (idc_información_velocidad_trama_constante > 0) {		
<b>reservado</b>	<b>4</b>	<b>bslbf</b>
<b>indicador_velocidad_trama</b>	<b>12</b>	<b>uimsbf</b>

Sintaxis	Nº de bits	Mnemónico
}		
}		
}		
}		

5 **[0060]** El documento de Hendry et al., "TRANSPORT STREAM FOR CARRIAGE OF VIDEO CODING EXTENSIONS [FLUJO DE TRANSPORTE PARA EL TRASLADO DE EXTENSIONES DE CODIFICACIÓN DE VÍDEO]" Solicitud Estadounidense N° 14/800.498, presentada el 15 de julio de 2015 , describe detalles sobre un diseño de MPEG-2 TS para el traslado de extensiones de la HEVC.

10 **[0061]** Las técnicas de esta divulgación pueden usarse para superar ciertos problemas con las técnicas existentes para la señalización de descriptores utilizados para la información de dependencia entre capas de flujos de bits de la HEVC en capas, tales como los que se exponen a continuación.

15 **[0062]** Esta divulgación reconoce una superposición de la funcionalidad del descriptor de jerarquía y del descriptor de extensión de jerarquía. En el texto de estudio de ISO/IEC 13818-1: 2013/PDAM7, hay dos descriptores para la señalización de la información de dependencia; a saber, el descriptor de jerarquía y el descriptor de extensión de jerarquía. Los dos descriptores tienen funcionalidad solapada, ya que ambos son capaces de describir el tipo de dependencia espacial, de calidad y de vista múltiple. En el borrador de L-HEVC TS, es posible que ambos descriptores estén presentes y asociados al mismo flujo elemental. Esto crearía confusión y añadiría complejidad innecesaria para la implementación de la norma.

20 **[0063]** Aunque el descriptor de jerarquía y el descriptor de extensión de jerarquía tienen funcionalidad solapada, ninguno de ellos, por sí mismo, puede usarse para describir todos los posibles tipos de dependencia entre capas en la SHVC y la MV-HEVC. El descriptor de jerarquía no es capaz de describir la dependencia de una imagen auxiliar, mientras que el descriptor de extensión de jerarquía no es capaz de describir la dependencia temporal.

25 **[0064]** Esta divulgación también reconoce que falta una descripción de cuándo estará presente el descriptor de extensión de jerarquía. En el texto de estudio de ISO/IEC 13818-1: 2013/PDAM7, la presencia del descriptor de jerarquía o del descriptor de extensión de jerarquía no es obligatoria. A continuación se describe cuándo estará presente el descriptor de jerarquía, pero no hay ninguna descripción de cuándo estará presente el descriptor de extensión de jerarquía:

30 *Cuando un programa de Rec. UIT-T H.222,0 | ISO/IEC 13818-1 incluye más de un subconjunto temporal de vídeo de HEVC, o más de un subflujo de bits de vídeo temporal HEVC y al menos un subconjunto temporal de vídeo de HEVC o al menos una subpartición de mejora de la HEVC, uno o más descriptores de jerarquía, según lo definido en 2.6.7, estará presente para todos los flujos elementales asociados a un tipo de flujo igual a 0x24, 0x25 o 0x27 a 0x2A. Los descriptores de jerarquía se utilizarán para indicar las dependencias de todos los subflujos de bits de vídeo temporal de la HEVC, los subconjuntos temporales de vídeo de la HEVC y las subparticiones de la HEVC.*

35 **[0065]** Además, esta divulgación reconoce que falta una descripción de la referencia de flujo elemental cuando no está presente ni el descriptor de jerarquía ni el descriptor de extensión de jerarquía. Para la señalización de los puntos operativos, se hace una referencia a un flujo elemental utilizando el valor de *índice\_capa\_jerarquía* que se señala en el descriptor de jerarquía y en el descriptor de extensión de jerarquía. Cuando ninguno entre el descriptor de jerarquía y el descriptor de extensión de jerarquía está presentes en un programa, no hay ninguna descripción de cómo se puede resolver la referencia al flujo elemental para la señalización de puntos operativos.

45 **[0066]** Esta divulgación también reconoce que hay problemas relacionados con la señalización de la información de perfil, grada y nivel (PTL). En el 9º JCT3V y la 18ª conferencia del JCT-VC, se acordó que la información de PTL ha de ser señalizada para cada capa en el flujo de bits de la HEVC en capas (es decir, la SHVC o la MV-HEVC). En consecuencia, toda la sintaxis y la semántica de la información de PTL están diseñadas para la capa. Por otro lado, en el borrador L-HEVC TS, la señalización de la información de PTL se señala para cada punto operativo, en lugar de para cada capa incluida en cada punto operativo. Dicha señalización no es correcta, ya que la señalización en el borrador de L-HEVC TS no especifica diferentes sintaxis y semántica para la información de PTL, sino que simplemente se refiere a la sintaxis y semántica de la información de PTL definidas en las especificaciones en borrador de la SHVC / la MV-HEVC.

55 **[0067]** Esta divulgación reconoce además que existe una ineficacia en la señalización de la información de PTL. Una capa en un flujo de bits de la HEVC en capas puede ser parte de uno o más puntos operativos. Teniendo en cuenta que la información de PTL debería asociarse a capas dentro de puntos operativos, es probable que las mismas estructuras de PTL se señalicen repetidamente si las estructuras de PTL se señalizan mediante el uso del descriptor de extensión de la HEVC o del punto operativo de la HEVC. Por lo tanto, la eficacia de la señalización de la información

de PTL debería mejorarse evitando repetir la señalización de la misma información.

**[0068]** Además, esta divulgación reconoce que falta cierta información en la señalización de puntos operativos. La señalización del punto operativo actual en el descriptor de extensión de HEVC o el descriptor de punto operativo de HEVC carece de información crítica, tal como la indicación del conjunto asociado de capas de salida y el esquema de partición. Estos dos trozos de información son críticos para determinar los parámetros HRD aplicables para los flujos elementales en cada punto operativo. Como se especifica en el Anexo F de la Rec. UIT-T H.265 | ISO/IEC 23008-2, el parámetro HRD aplicable para una partición está determinado por el valor de *idx\_hrd\_bsp*, que está indizado por un índice para el conjunto de capas de salida de destino (*OlsIdxDestino*), un esquema de partición (*PsIdxDestino*), el máximo identificador temporal (*MáximoldT*), el índice de planificación de entrega (*PlanifSelComblDx*), así como un índice para la partición (*idxPartición*). Para el transporte del flujo de bits de la HEVC en capas, el *MáximoldT* se asigna en función del Identificador temporal aplicable / máximo de un punto operativo, *PlanifSelComblDx* se asigna en función del *IdxSelPlanif* descrito en la temporización de la HEVC y el descriptor HRD, y el *idxPartición* se asigna en función del índice de la subpartición. Sin embargo, no hay ningún valor señalado / obtenido en ningún descriptor dentro del texto de estudio de ISO/IEC 13818-1: 2013/PDAM7 que se pueda usar para asignar *IdxOlsDestino* e *IdxPsDestino*.

**[0069]** Esta divulgación también reconoce que existe la posibilidad de información de perfil faltante, de grada faltante y/o de nivel faltante para un programa. De acuerdo a la descripción de la agrupación de flujo elemental en la subcláusula 2.17.4 del borrador de L-HEVC TS, la presencia del descriptor de extensión de la HEVC y del descriptor de punto operativo de la HEVC no es obligatoria. Cuando ni el descriptor de extensión de la HEVC ni el descriptor de punto operativo de la HEVC están presentes, no hay ninguna información de PTL asociada a flujos elementales en el programa. La presencia de información de PTL es importante por dos motivos. Primero, la información de PTL es útil para propósitos de negociación del sistema. Cuando la información de PTL no está disponible en el nivel de transporte, la entidad del sistema (por ejemplo, la caja central inteligente en el medio del sistema de suministro) se ve obligada a descender al nivel del códec (es decir, la entidad del sistema debe utilizar y/o gastar más recursos para determinar la información de PTL desde el nivel de códec que los recursos requeridos para determinar la información de PTL desde el nivel de transporte), lo cual es una carga. En segundo lugar, la información de PTL puede ser necesaria para un modelo de almacén temporal cuando no hay ningún parámetro HRD presente.

**[0070]** A continuación se describen ejemplos de acuerdo a la divulgación. Las técnicas ejemplares pueden implementarse juntas o por separado. En algunos ejemplos, las técnicas ejemplares pueden abordar los problemas descritos anteriormente. Sin embargo, no es necesario abordar los problemas descritos anteriormente. Por ejemplo, las técnicas descritas en esta divulgación no deberían considerarse limitadas a abordar los problemas descritos anteriormente, o proporcionar necesariamente las ventajas descritas en esta divulgación. Los problemas descritos anteriormente y las ventajas potenciales se proporcionan meramente como contexto y para ayudar a la comprensión, y no deberían considerarse como un requisito.

**[0071]** A continuación se proporciona un resumen de ciertos procedimientos de esta divulgación, con una implementación detallada de algunas técnicas ejemplares proporcionadas más adelante. Algunas de estas técnicas ejemplares se pueden aplicar de forma independiente y algunas de ellas se pueden aplicar en combinación.

**[0072]** En algunos ejemplos, el descriptor existente de la extensión de la HEVC y el descriptor de punto operativo de la HEVC se pueden combinar en un solo descriptor (es decir, un descriptor único) y este descriptor único puede reutilizar el nombre "descriptor de punto operativo de la HEVC". Este descriptor de punto operativo de la HEVC puede tener los siguientes componentes:

- a. Una lista de estructuras de PTL. Las estructuras de PTL en la lista serán referidas y asociadas a capas del punto operativo mediante índices.
- b. Una lista de puntos operativos. Cada punto operativo incluye la siguiente información:
  - i. El conjunto asociado de capas de salida
  - ii. El esquema de división asociado
  - iii. La subcapa temporal más alta
  - iv. Lista de flujos elementales (es decir, subpartición) que conforman el punto operativo
  - v. El número de capas de salida
  - vi. Correlación de capa contenida en cada flujo elemental del punto operativo con PTL
  - vii. Información de velocidad de trama

**[0073]** Cuando la dependencia entre flujos elementales dentro de un programa de Rec. H.222.0 | ISO/CEI 13818-1 del ITU-T está disponible (por ejemplo, señalizada en el descriptor de jerarquía o descriptor de extensión de jerarquía), la lista de flujos elementales (es decir, la subpartición) que conforman un punto operativo puede incluir solo el número mínimo de flujos elementales. Al utilizar la información de dependencia disponible, se puede completar la lista de flujos elementales que conforman el punto operativo.

**[0074]** Cuando la dependencia entre flujos elementales dentro de un programa de Rec. H.222.0 | ISO/IEC 13818-1



de la ITU-T está disponible (por ejemplo, señalizada en el descriptor de jerarquía o descriptor de extensión de jerarquía), se puede requerir que la lista de flujos elementales (es decir, la subpartición) que conforman un punto operativo incluya la lista completa de flujos elementales que conforman el punto operativo.

5 **[0075]** En un ejemplo, en lugar de señalar el número de capas de salida en un punto operativo, para cada capa incluida en el punto operativo, se señala un indicador para indicar si la capa es o no una capa de salida. Alternativamente, se puede señalar tanto el número de capas de salida como la indicación del indicador (que indica si una capa respectiva es una capa de salida) para cada capa.

10 **[0076]** Cuando hay más de un descriptor de punto operativo de la HEVC presente para el programa de la Rec. H.222,0 | ISO/IEC 13818-1 de la ITU-T, se puede aplicar lo siguiente. Primero, la lista de información de PTL señalizada en el (n+1)-ésimo descriptor de punto operativo de la HEVC, donde n comienza desde 1, puede ser la continuación de la lista de información de PTL señalizada en el n-ésimo descriptor de punto operativo de la HEVC. En segundo lugar, la lista de puntos operativos señalizados en el (n+1)-ésimo descriptor de punto operativo de la HEVC, donde n comienza en 1, puede ser la continuación de la lista de puntos operativos señalizados en el n-ésimo descriptor de punto operativo de la HEVC. En otras palabras, la información de dos o más descriptores consecutivos de puntos operativos de la HEVC, como se señalizan, pueden concatenarse para formar un único descriptor. Es decir, se puede señalar información similar en dos o más descriptores distintos de puntos operativos de la HEVC, y esta información se puede concatenar como si toda la información estuviera señalizada en un solo descriptor.

20 **[0077]** El descriptor de punto operativo de la HEVC descrito en este documento puede ser un descriptor a nivel de programa (es decir, señalado junto con otros descriptores que vienen inmediatamente después del campo del elemento sintáctico `longitud_información_programa` en la tabla de mapas de programa). Alternativamente, el descriptor de punto operativo de la HEVC puede ser un descriptor elemental de nivel de flujo (es decir, señalado junto con otros descriptores que vienen después del campo del elemento sintáctico `longitud_información_ES` en la tabla de mapas de programa).

25 **[0078]** La presencia del descriptor de punto operativo de la HEVC puede ser obligatoria cuando un programa tiene uno o más flujos elementales con `tipo_flujo` igual a 0x27, 0x28, 0x29 o 0x2A. Del mismo modo, la presencia del descriptor de vídeo de la HEVC puede ser obligatoria para cada flujo elemental con `tipo_flujo` igual a 0x24 y 0x25.

30 **[0079]** En algunos ejemplos, el descriptor de extensión de jerarquía puede modificarse para prestar soporte a la descripción de la dependencia/mejora temporal. Con este fin, uno de los valores reservados en la Tabla amd7-4 - Semántica de bits de dimensión de extensión en el borrador de L-HEVC TS se asigna para mejora temporal.

35 **[0080]** En algunos ejemplos, el descriptor de jerarquía se modifica para prestar soporte a la mejora de capa auxiliar. Con este fin, puede valer lo siguiente:

40 a. Utilizar uno de los indicadores reservados en la tabla de sintaxis de descriptor\_jerarquía para indicar la mejora auxiliar.

b. Asignar uno de los valores reservados en la Tabla 2-50: valores de campos de tipo\_jerarquía para indicar una mejora auxiliar.

45 **[0081]** En algunos ejemplos, para evitar el uso solapado del descriptor de jerarquía y el descriptor de extensión de jerarquía, se puede utilizar lo siguiente:

50 a. Para cada flujo elemental que lleva la imagen codificada por HEVC, un descriptor (y, en algunos ejemplos, exactamente uno), bien el descriptor de jerarquía, o bien el descriptor de extensión de jerarquía, puede estar (y, en algunos ejemplos, estará) presente y asociado al flujo elemental.

b. Para cada flujo elemental que contiene imágenes con `idCapa` igual a 0, un descriptor de jerarquía puede estar (y, en algunos ejemplos, estará) presente y asociado al flujo elemental.

55 c. Para cada flujo elemental que no contiene imágenes con `idCapa` igual a 0 y que contiene imágenes con `idCapa` no igual a 0, un descriptor de extensión de jerarquía puede estar (y, en algunos ejemplos, estará (es decir, debe estar)) presente y asociado al flujo elemental.

60 **[0082]** 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 a 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 para ser decodificados posteriormente por un dispositivo de destino 14. En particular, el dispositivo de origen 12 proporciona los datos de vídeo al dispositivo de destino 14 a través de 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, ordenadores portátiles), ordenadores de tableta, decodificadores, equipos telefónicos manuales tales como los llamados teléfonos

"inteligentes", tabletas, televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivos de transmisión continua de vídeo o similares. En algunos casos, el dispositivo de origen 12 y el dispositivo de destino 14 pueden estar equipados para la comunicación inalámbrica.

5 **[0083]** El dispositivo de destino 14 puede recibir los datos de vídeo codificados para ser decodificados 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 para permitir que el dispositivo de origen 12 transmita datos de vídeo codificados directamente al dispositivo de destino 14 en tiempo real. Los datos de vídeo codificados pueden modularse de acuerdo a una norma de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitirse al dispositivo de destino 14. El medio de comunicación puede comprender cualquier medio de comunicación inalámbrico o por cable, tal como un espectro de radiofrecuencia (RF) o una o más líneas 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 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 al dispositivo de destino 14.

20 **[0084]** En algunos ejemplos, los datos codificados pueden emitirse desde la interfaz de salida 22 a un dispositivo de almacenamiento. De manera similar, se puede acceder a datos codificados desde el dispositivo de almacenamiento mediante la interfaz de entrada. El dispositivo de almacenamiento puede incluir cualquiera entre varios medios de almacenamiento de datos distribuidos o de acceso local, tales como un disco duro, discos Blu-ray, DVD, CD-ROM, memoria flash, memoria volátil o no volátil, u otros medios cualesquiera de almacenamiento digital adecuados para almacenar datos de vídeo codificados. En un ejemplo adicional, el dispositivo de almacenamiento puede corresponder a un servidor de ficheros u otro dispositivo de almacenamiento intermedio que pueda almacenar el vídeo codificado generado por el dispositivo de origen 12. El dispositivo de destino 14 puede acceder a los datos de vídeo almacenados del dispositivo de almacenamiento mediante transmisión continua 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 ejemplares de ficheros incluyen un servidor de la Red (por ejemplo, para una sede de la Red), un servidor del FTP, dispositivos de almacenamiento conectados a red (NAS) o una unidad de disco local. El dispositivo de destino 14 puede acceder a los datos de vídeo codificados a través de cualquier conexión de datos estándar, incluida una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión de Wi-Fi), una conexión por cable (por ejemplo, DSL, módem de cable, etc.), o una combinación de ambos que sea adecuada para acceder a datos de vídeo codificados almacenados en un servidor de ficheros. La transmisión de datos de vídeo codificados desde el dispositivo de almacenamiento puede ser una transmisión continua, una transmisión de descarga o una combinación de las mismas.

40 **[0085]** Las técnicas de esta divulgación no se limitan necesariamente a las aplicaciones o configuraciones inalámbricas. Las técnicas pueden aplicarse a la codificación de vídeo en apoyo de cualquiera entre varias aplicaciones de multimedios, tales como difusiones de televisión por aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones continuas de vídeo por Internet, tales como la transmisión continua dinámica adaptativa por el HTTP (DASH), vídeo digital que se codifica en un medio de almacenamiento de datos, decodificación de vídeo digital almacenado en un medio de almacenamiento de datos u otras aplicaciones. En algunos ejemplos, el sistema 10 puede configurarse para prestar soporte a la transmisión de vídeo unidireccional o bidireccional, para prestar soporte a aplicaciones tales como la transmisión continua de vídeo, la reproducción de vídeo, la difusión de vídeo y/o la videotelefonía.

50 **[0086]** En el ejemplo de la figura 1, el dispositivo de origen 12 incluye el origen de vídeo 18, el codificador de vídeo 20, el multiplexor 21 y la interfaz de salida 22. El dispositivo de destino 14 incluye la interfaz de entrada 28, el demultiplexor 29, el decodificador de vídeo 30 y el dispositivo de visualización 32. De acuerdo a esta divulgación, el multiplexor 21 del dispositivo de origen 12 puede configurarse para aplicar las técnicas para transportar datos de vídeo codificados de acuerdo a las extensiones de una norma de codificación de vídeo, mientras que el demultiplexor 29 puede recibir dichos datos para procesar y remitir los datos de vídeo procesados, por ejemplo, al 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 externo de vídeo 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.

60 **[0087]** El sistema ilustrado 10 de la figura 1 es simplemente un ejemplo. Las técnicas para transportar datos de vídeo codificados de acuerdo a 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 se pueden realizar mediante un codificador/decodificador de vídeo, generalmente mencionado como "CÓDEC". Además, las técnicas de esta divulgación también pueden ser realizadas por un preprocesador de vídeo. El dispositivo de origen 12 y el dispositivo de destino 14 son simplemente ejemplos de tales 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 una manera esencialmente simétrica, de modo que cada uno de los

dispositivos 12, 14 incluya componentes de codificación y 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 transmisión continua de vídeo, reproducción de vídeo, difusión de vídeo o videotelefonía.

5 **[0088]** El origen de vídeo 18 del dispositivo de origen 12 puede incluir un dispositivo de captura de vídeo, tal como una cámara de vídeo, un archivo de vídeo que contenga vídeo capturado previamente y/o una interfaz de alimentación de vídeo para recibir vídeo desde un proveedor de contenido de vídeo. Como una alternativa adicional, el origen de vídeo 18 puede generar datos basados en gráficos de ordenador, como el vídeo de origen, o una combinación de vídeo en vivo, vídeo archivado y vídeo generado por ordenador. En algunos casos, si el origen de vídeo 18 es una  
10 cámara de vídeo, el dispositivo de origen 12 y el dispositivo de destino 14 pueden formar los llamados teléfonos con cámara o videoteléfonos. Sin embargo, como se ha mencionado anteriormente, las técnicas descritas en esta divulgación pueden ser aplicables a la codificación de vídeo en general, y pueden aplicarse a aplicaciones inalámbricas y/o cableadas. En cada caso, el vídeo capturado, precapturado o generado por ordenador puede estar codificado por el codificador de vídeo 20. La información de vídeo codificada puede luego ser emitida por la interfaz de salida 22 a  
15 un medio legible por ordenador 16.

**[0089]** El medio legible por ordenador 16 puede incluir medios transitorios, tales como una difusión inalámbrica o una transmisión de red por cable, o medios de almacenamiento (es decir, medios de almacenamiento no transitorio), tales como un disco duro, una unidad flash, un disco compacto, un disco de vídeo digital, un disco Blu-ray u otro medio legible por ordenador. En algunos ejemplos, un servidor de red (no mostrado) 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 de red. De manera similar, un dispositivo informático de una utilidad de producción media, tal como una utilidad de estampado de discos, puede recibir datos de vídeo codificados desde el dispositivo de origen 12 y producir un disco que contenga los datos de vídeo codificados. Por lo tanto, se puede entender que el medio legible por ordenador 16 incluye uno o más medios legibles por ordenador de varias formas, en varios ejemplos.  
20  
25

**[0090]** La interfaz de entrada 28 del dispositivo de destino 14 recibe información desde un medio legible por ordenador 16. La información del medio legible por ordenador 16 puede incluir información sintáctica definida por el codificador de vídeo 20, que también es usada por el decodificador de vídeo 30, que incluye elementos sintácticos que describen características y/o procesamiento de bloques y otras unidades codificadas, por ejemplo, los GOP. El dispositivo de visualización 32 muestra los datos de vídeo decodificados a un usuario, y puede comprender cualquiera entre varios 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.  
30  
35

**[0091]** El codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar de acuerdo a una norma de codificación de vídeo, tal como la norma de codificación de vídeo de alta eficacia (HEVC), actualmente en desarrollo, y pueden cumplir con el modelo de prueba (HM) de la HEVC. Alternativamente, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden funcionar de acuerdo a otras normas patentadas o industriales, tales como la norma ITU-T H.264, conocida alternativamente como MPEG-4, Parte 10, Codificación de vídeo avanzada (AVC), o extensiones de tales normas. Las técnicas de esta divulgación, sin embargo, no están limitadas a ninguna norma de codificación particular. Otros ejemplos de normas de codificación de vídeo incluyen MPEG-2 y UIT-T H.263. Aunque no se muestra en la figura 1, en algunos aspectos, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden estar integrados con un codificador y decodificador de audio, y pueden incluir unidades MUX-DEMUX adecuadas, u otro hardware y software, para gestionar la codificación tanto de audio como de vídeo en un flujo común de datos o flujos de datos distintos. Si corresponde, las unidades MUX-DEMUX pueden cumplir con el protocolo multiplexor ITU H.223 u otros protocolos, tales como el protocolo de datagramas de usuario (UDP).  
40  
45

**[0092]** La norma UIT-T H.264/MPEG-4 (AVC) fue formulada por el Grupo de expertos en codificación de vídeo (VCEG) del UIT-T junto con el Grupo de expertos en imágenes en movimiento (MPEG) de ISO/IEC, como el producto de una asociación colectiva conocida como el Equipo de vídeo Conjunto (JVT). En algunos aspectos, las técnicas descritas en esta divulgación pueden aplicarse a dispositivos que generalmente cumplen con la norma H.264. La norma H.264 se describe en la Recomendación UIT-T H.264, Codificación avanzada de vídeo para servicios audiovisuales genéricos, realizada por el Grupo de Estudio de la UIT-T, con fecha de marzo de 2005, a la que se puede hacer referencia aquí como la norma H.264, o la especificación H.264, o la norma o especificación H.264/AVC. El equipo de vídeo conjunto (JVT) continúa trabajando en las extensiones de H.264/MPEG-4 AVC.  
50  
55

**[0093]** El codificador de vídeo 20 y el decodificador de vídeo 30 pueden implementarse, cada uno, como cualquiera entre varios circuitos codificadores adecuados, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), formaciones de compuertas programables en el terreno (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 no transitorio legible por ordenador adecuado 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 puede incluirse en uno o más codificadores o decodificadores, cualquiera de los cuales puede integrarse como parte de un codificador/decodificador combinado (CÓDEC) en un dispositivo respectivo.  
60  
65

**[0094]** El JCT-VC ha desarrollado la norma HEVC y continúa trabajando en las extensiones de la norma HEVC. Los esfuerzos de estandarización de la HEVC se basan en un modelo en evolución de un dispositivo de codificación de vídeo denominado Modelo de prueba de la HEVC (HM). El HM supone varias capacidades adicionales de los dispositivos de codificación de vídeo en relación con los dispositivos existentes, de acuerdo, por ejemplo, a ITU-T H.264/AVC. Por ejemplo, mientras que H.264 proporciona nueve modalidades de codificación intrapredicción, el HM puede proporcionar hasta treinta y tres (y tal vez treinta y cinco) modalidades de codificación intrapredicción.

**[0095]** En general, el modelo de trabajo del HM describe que una trama o imagen de vídeo se puede dividir en una secuencia de bloques arbolados o unidades máximas de codificación (LCU) (también denominadas "unidades de árbol de codificación") que incluyen muestras tanto de luma como de croma. Los datos sintácticos dentro de un flujo de bits pueden definir un tamaño para la LCU, que es la unidad de codificación máxima en términos del número de píxeles. Un fragmento incluye varios bloques arbolados consecutivos en orden de codificación. Un trama o imagen de vídeo se puede dividir en una o más fragmentos. Cada bloque arbolado se puede dividir en unidades de codificación (CU) de acuerdo a un árbol cuádruple. En general, una estructura de datos de árbol cuádruple incluye un nodo por CU, con un nodo raíz correspondiente al bloque arbolado. Si una CU se divide en cuatro subCU, el nodo correspondiente a la CU incluye cuatro nodos de hoja, cada uno de los cuales corresponde a una de las subCU.

**[0096]** Cada nodo de la estructura de datos de árbol cuádruple puede proporcionar datos sintácticos para la CU correspondiente. Por ejemplo, un nodo en el árbol cuádruple puede incluir un indicador de división, que indica si la CU correspondiente al nodo se divide en subCU. Los elementos sintácticos para una CU pueden definirse de forma recursiva y pueden depender de si la CU se divide en subCU. Si una CU no se divide más, se denomina hoja-CU. En esta descripción, cuatro subCU de una hoja-CU también se conocerán como hojas-CU incluso si no hay ninguna división explícita de la hoja-CU original. Por ejemplo, si una CU de tamaño 16x16 no se divide más, las cuatro subCU de tamaño 8x8 también se denominarán hojas-CU, aunque la CU de tamaño 16x16 nunca se dividió.

**[0097]** Una CU tiene un propósito similar a un macrobloque de la norma H.264, excepto que una CU no tiene una distinción de tamaño. Por ejemplo, un bloque arbolado puede dividirse en cuatro nodos secundarios (también denominados subCU), y cada nodo secundario puede a su vez ser un nodo primario y dividirse en otros cuatro nodos secundarios. Un nodo secundario final, no dividido, denominado nodo hoja del árbol cuádruple, comprende un nodo de codificación, también denominado hoja-CU. Los datos sintácticos asociados a un flujo de bits codificado pueden definir un número máximo de veces que un bloque arbolado puede dividirse, lo que se conoce como una profundidad máxima de CU, y también puede definir un tamaño mínimo de los nodos de codificación. Por consiguiente, un flujo de bits también puede definir una unidad mínima de codificación (SCU). Esta divulgación utiliza el término "bloque" para referirse a cualquiera entre una CU, PU o TU, en el contexto de la HEVC, o a estructuras de datos similares en el contexto de otras normas (por ejemplo, macrobloques y subbloques de los mismos en H.264/AVC).

**[0098]** Una CU incluye un nodo de codificación y unidades de predicción (PU) y unidades de transformación (TU) asociadas al nodo de codificación. Un tamaño de la CU corresponde a un tamaño del nodo de codificación y debe tener forma cuadrada. El tamaño de la CU puede variar desde 8x8 píxeles hasta el tamaño del bloque arbolado, con un máximo de 64x64 píxeles o más. Cada CU puede contener una o más PU y una o más TU. Los datos sintácticos asociados a una CU pueden describir, por ejemplo, la partición de la CU en una o más PU. Las modalidades de partición pueden diferir entre si la CU está codificada en modalidad de omisión o directa, codificada en modalidad de intrapredicción o codificada en modalidad de interpredicción. Las PU se pueden dividir para que no tengan forma cuadrada. Los datos sintácticos asociados a una CU también pueden describir, por ejemplo, la partición de la CU en una o más TU de acuerdo a un árbol cuádruple. Una TU puede tener forma cuadrada o no cuadrada (por ejemplo, rectangular).

**[0099]** La norma HEVC admite transformaciones según las TU, que pueden ser diferentes para diferentes CU. Las TU suelen tener un tamaño basado en el tamaño de las PU dentro de una CU determinada definida para una LCU particionada, aunque este no siempre sea el caso. Las TU son típicamente del mismo tamaño, o más pequeñas, que las PU. En algunos ejemplos, las muestras residuales correspondientes a una CU pueden subdividirse en unidades más pequeñas utilizando una estructura de árbol cuádruple conocida como "árbol cuádruple residual" (RQT). Los nodos de hoja del RQT pueden denominarse unidades de transformación (TU). Los valores de diferencia de píxeles asociados a las TU pueden transformarse para producir coeficientes de transformación, que pueden cuantizarse.

**[0100]** El codificador de vídeo 20 puede usar la partición de árbol cuádruple para descomponer los bloques residuales de luma, Cb y Cr de una CU en uno o más bloques de transformación de luma, Cb y Cr. Un bloque de transformación puede ser un bloque rectangular de muestras en el que se aplica la misma 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 utilizadas para transformar las muestras de bloque de transformación. Por lo tanto, cada TU de una CU puede estar asociada a un bloque de transformación de luma, un bloque de transformación de Cb y un bloque de transformación de Cr. El bloque de transformación de luma asociado a la TU puede ser un subbloque del bloque residual de luma de la CU. El bloque de transformación de Cb puede ser un subbloque del bloque residual de Cb de la CU. El bloque de transformación de Cr puede ser un subbloque del bloque residual de Cr de la CU. En imágenes monocromas o imágenes que tienen tres

planos de color distintos, una TU puede comprender un solo bloque de transformación y estructuras sintácticas utilizadas para transformar las muestras del bloque de transformación.

**[0101]** Una hoja-CU puede incluir una o más unidades de predicción (PU). En general, una PU representa un área espacial correspondiente a la totalidad o a una parte de la CU correspondiente, y puede incluir datos para recuperar una muestra de referencia para la PU. Por otra parte, una PU incluye datos relacionados con la predicción. Por ejemplo, cuando la PU está codificada en intramodalidad, los datos para la PU pueden incluirse en un árbol cuádruple residual (RQT), que puede incluir datos que describen una modalidad de intrapredicción para una TU correspondiente a la PU. Como otro ejemplo, cuando la PU está codificada en intermodalidad, la PU puede incluir datos que definen uno o más vectores de movimiento para la PU. Los datos que definen el vector de movimiento para una PU pueden describir, por ejemplo, un componente horizontal del vector de movimiento, un componente vertical del vector de movimiento, una resolución para el vector de movimiento (por ejemplo, precisión de un cuarto de píxel o un octavo de píxel), una imagen de referencia a la que señala el vector de movimiento y/o una lista de imágenes de referencia (por ejemplo, Lista 0, Lista 1 o Lista C) para el vector de movimiento.

**[0102]** Una hoja-CU que tiene una o más PU también puede incluir una o más unidades de transformación (TU). Las unidades de transformación se pueden especificar utilizando un RQT (también mencionado como una estructura de árbol cuádruple de TU), como se ha expuesto anteriormente. Por ejemplo, un indicador de división puede indicar si una hoja-CU se divide en cuatro unidades de transformación. Luego, cada unidad de transformación se puede dividir en subTU adicionales. Cuando una TU no se divide más, puede denominarse una hoja-TU. Generalmente, para la intracodificación, todas las hojas-TU que pertenecen a una hoja-CU comparten la misma modalidad de intrapredicción. Es decir, la misma modalidad de intrapredicción se aplica generalmente para calcular los valores predichos para todas las TU de una hoja-CU. Para la intracodificación, un codificador de vídeo puede calcular un valor residual para cada hoja-TU utilizando la modalidad de intrapredicción, como una diferencia entre la parte de la CU correspondiente a la TU y el bloque original. Una TU no se limita necesariamente al tamaño de una PU. Por lo tanto, las TU pueden ser más grandes o más pequeñas que una PU. Para la intracodificación, una PU puede cosituarse con una hoja-TU correspondiente para la misma CU. En algunos ejemplos, el tamaño máximo de una hoja-TU puede corresponder al tamaño de la hoja-CU correspondiente.

**[0103]** Además, las TU de las hojas-CU también pueden asociarse a las respectivas estructuras de datos de árbol cuádruple, denominadas árboles cuádruples residuales (RQT). Es decir, una hoja-CU puede incluir un árbol cuádruple que indica cómo se divide la hoja-CU en las TU. El nodo raíz de un árbol cuádruple de TU generalmente corresponde a una hoja-CU, mientras que el nodo raíz de un árbol cuádruple de CU generalmente corresponde a un bloque arbolado (o LCU). Las TU del RQT que no están divididas se denominan hojas-TU. En general, esta divulgación utiliza los términos CU y TU para referirse, respectivamente, a hoja-CU y hoja-TU, a menos que se indique lo contrario.

**[0104]** Una secuencia de vídeo normalmente incluye una serie de tramas o imágenes de vídeo. Un grupo de imágenes (GOP) generalmente comprende una serie de una o más de las imágenes de vídeo. Un GOP puede incluir datos sintácticos en un encabezado del GOP, un encabezado de una o más de las imágenes, o en otro lugar, que describen una serie de imágenes incluidas en el GOP. Cada fragmento de una imagen puede incluir datos sintácticos de fragmento que describen una modalidad de codificación para el respectivo fragmento. El codificador de vídeo generalmente opera sobre bloques de vídeo dentro de fragmentos de vídeo individuales, para codificar los datos de vídeo. Un bloque de vídeo puede corresponder a un nodo de codificación dentro de una CU. Los bloques de vídeo pueden tener tamaños fijos o variables, y pueden diferir en tamaño según una norma de codificación especificada.

**[0105]** Como ejemplo, el HM presta soporte a la predicción en varios tamaños de PU. Suponiendo que el tamaño de una CU en particular es  $2N \times 2N$ , el HM presta soporte a la intrapredicción en los tamaños de PU de  $2N \times 2N$  o  $N \times N$ , y a la interpredicción en los tamaños de PU simétricos de  $2N \times 2N$ ,  $2N \times N$ ,  $N \times 2N$  o  $N \times N$ . El HM también presta soporte a la partición asimétrica para la interpredicción en los tamaños de PU de  $2N \times nU$ ,  $2N \times nD$ ,  $nL \times 2N$  y  $nR \times 2N$ . En la partición asimétrica, una dirección de una CU no se divide, mientras que la otra dirección se divide en un 25% y un 75%. La parte de la CU correspondiente a la partición del 25% se indica con una "n" seguida de una indicación de "Arriba", "Abajo", "Izquierda" o "Derecha". Así, por ejemplo, " $2N \times nU$ " se refiere a una CU de tamaño  $2N \times 2N$  que está dividida horizontalmente, con una PU de tamaño  $2N \times 0,5N$  en la parte superior y una PU de tamaño  $2N \times 1,5N$  en la parte inferior.

**[0106]** En esta descripción, " $N \times N$ " y " $N$  por  $N$ " se pueden usar indistintamente para referirse a las dimensiones en píxeles de un bloque de vídeo, en términos de dimensiones verticales y horizontales, por ejemplo,  $16 \times 16$  píxeles o 16 por 16 píxeles. En general, un bloque de tamaño  $16 \times 16$  tendrá 16 píxeles en una dirección vertical ( $y = 16$ ) y 16 píxeles en una dirección horizontal ( $x = 16$ ). Del mismo modo, un bloque de tamaño  $N \times N$  generalmente tiene  $N$  píxeles en una dirección vertical y  $N$  píxeles en una dirección horizontal, donde  $N$  representa un valor entero no negativo. Los píxeles en un bloque se pueden organizar en filas y columnas. Además, los bloques no necesariamente han de tener el mismo número de píxeles en la dirección horizontal que en la dirección vertical. Por ejemplo, los bloques pueden comprender  $N \times M$  píxeles, donde  $M$  no es necesariamente igual a  $N$ .

**[0107]** Después de la codificación intrapredicativa o interpredicativa, usando las PU de una CU, el codificador de vídeo puede calcular datos residuales para las TU de la CU. Las PU pueden comprender datos sintácticos que describen un procedimiento o modalidad de generar datos de píxeles predictivos en el dominio espacial (también denominado

dominio de píxeles) y las TU pueden comprender coeficientes en el dominio de transformación después de la aplicación de una transformación, por ejemplo, una transformación de coseno discreta (DCT), una transformación entera, una transformación por ondículas o una transformación conceptualmente similar, a datos de vídeo residuales. Los datos residuales pueden corresponder a las diferencias de píxeles entre los píxeles de la imagen no codificada y los valores de predicción correspondientes a las PU. El codificador de vídeo 20 puede formar las TU, incluidos los datos residuales para la CU, y luego transformar las TU para producir coeficientes de transformación para la CU.

**[0108]** Después de cualquier transformación para producir coeficientes de transformación, el codificador de vídeo 20 puede realizar la cuantización de los coeficientes de transformación. La cuantización generalmente se refiere a un proceso en el que los coeficientes de transformación se cuantizan para reducir posiblemente la cantidad de datos utilizados para representar los coeficientes, proporcionando una compresión adicional. El proceso de cuantización puede reducir la profundidad de bits asociada a algunos de, o todos, los coeficientes. Por ejemplo, un valor de  $n$  bits se puede redondear a la baja hasta un valor de  $m$  bits durante la cuantización, donde  $n$  es mayor que  $m$ .

**[0109]** Después de la cuantización, el codificador de vídeo puede recorrer los coeficientes de transformación, produciendo un vector unidimensional a partir de la matriz bidimensional que incluye los coeficientes de transformación cuantizados. El recorrido puede diseñarse para colocar coeficientes de mayor energía (y, por lo tanto, de menor frecuencia) al frente de la formación y para colocar coeficientes de menor energía (y, por lo tanto, de mayor frecuencia) en la parte posterior de la formación. En algunos ejemplos, el codificador de vídeo 20 puede utilizar un orden de recorrido predefinido para recorrer los coeficientes de transformación cuantizados, para producir un vector serializado que pueda codificarse por entropía. En otros ejemplos, el codificador de vídeo 20 puede realizar un recorrido adaptativo. Después de recorrer los coeficientes de transformación cuantizados para formar un vector unidimensional, el codificador de vídeo 20 puede codificar por entropía el vector unidimensional, por ejemplo, de acuerdo a la codificación de longitud variable adaptativa al contexto (CAVLC), la codificación aritmética binaria adaptativa al contexto (CABAC), la codificación de aritmética binaria adaptativa al contexto basada en la sintaxis (SBAC), la codificación por Entropía de partición de intervalos de probabilidad (PIPE) u otra metodología de codificación por entropía. El codificador de vídeo 20 también puede codificar por entropía los elementos sintácticos asociados a los datos de vídeo codificados para su uso por el decodificador de vídeo 30 en la decodificación de los datos de vídeo.

**[0110]** Para realizar la CABAC, el codificador de vídeo 20 puede asignar un contexto dentro de un modelo de contexto a un símbolo a transmitir. El contexto puede referirse a, por ejemplo, si los valores vecinos del símbolo son distintos de cero o no. Para realizar la CAVLC, el codificador de vídeo 20 puede seleccionar un código de longitud variable para un símbolo a transmitir. Las palabras de código en la VLC pueden construirse de manera tal que los códigos relativamente más cortos correspondan a símbolos más probables, mientras que los códigos más largos correspondan a símbolos menos probables. De esta manera, el uso de la VLC puede lograr un ahorro de bits sobre, por ejemplo, el uso de palabras de código de igual longitud para cada símbolo a transmitir. La determinación de probabilidad se puede basar en un contexto asignado al símbolo.

**[0111]** Esta divulgación describe técnicas para el traslado de flujos de bits de extensión de la HEVC. Es decir, de acuerdo a las técnicas de esta divulgación, el multiplexor 21 y/o el demultiplexor 29 pueden configurarse para transportar datos de vídeo (es decir, enviar o recibir datos de vídeo) que se codifican de acuerdo a una extensión de una norma de codificación de vídeo, tal como la HEVC, una extensión de la norma HEVC (por ejemplo, la SHVC o la MV-HEVC), u otras normas de codificación de vídeo aún no desarrolladas. En general, el multiplexor 21 puede encapsular datos de vídeo codificados para formar un flujo de bits, por ejemplo, esencialmente de acuerdo a los Sistemas MPEG-2 y las técnicas de esta divulgación, mientras que el demultiplexor 29 puede recibir y desencapsular datos encapsulados, por ejemplo, datos de vídeo codificados de acuerdo a una extensión de una norma de codificación de vídeo.

**[0112]** En algunos ejemplos, el multiplexor 21 y el demultiplexor 29 se pueden configurar para codificar una versión revisada del descriptor de punto operativo de la Tabla Amd7-1, como se ha expuesto anteriormente. La tabla a continuación, etiquetada como "Tabla propuesta Amd7-1 - descriptor de punto operativo de la HEVC", representa un ejemplo de modificaciones a la tabla actual Amd7-1. El descriptor de punto operativo HEVC descrito a continuación puede usarse para realizar ciertas técnicas de esta divulgación. El texto en cursiva en la tabla a continuación enfatiza las adiciones relativas a la Tabla actual Amd7-1, como se muestra arriba.

**Tabla propuesta Amd7-1 - Descriptor de punto operativo HEVC**

Sintaxis	Nº de bits	Mnemónico
descriptor_punto_operativo_HEVC () {		
<b>etiqueta_descriptor</b>	<b>8</b>	<b>uimsbf</b>
<b>longitud_descriptor</b>	<b>8</b>	<b>uimsbf</b>
<b>reservado</b>	<b>1</b>	<b>uimsbf</b>
<b>núm_perfil_grada_nivel</b>	<b>7</b>	<b>uimsbf</b>

Sintaxis	Nº de bits	Mnemónico
<i>para</i> ( <i>i</i> = 0; <i>i</i> < <i>núm_perfil_grada_nivel</i> ; <i>i</i> ++,		
<i>ptlldx</i> ++ ) {	2	uimsbf
<i>espacio_perfil</i> [ <i>ptlldx</i> ]	1	bslbf
<i>indicador_grada</i> [ <i>ptlldx</i> ]	5	uimsbf
<i>idc_perfil</i> [ <i>ptlldx</i> ]	32	bslbf
<i>indicación_compatibilidad_perfil</i> [ <i>ptlldx</i> ]	1	bslbf
<i>indicador_origen_progresivo</i> [ <i>ptlldx</i> ]	1	bslbf
<i>indicador_origen_entrelazado</i> [ <i>ptlldx</i> ]	1	bslbf
<i>indicador_restricción_no_empaquetada</i> [ <i>ptlldx</i> ]	1	bslbf
<i>indicador_restricción_trama_solamente</i> [ <i>ptlldx</i> ]	44	bslbf
<i>reservado_cero_44bits</i> [ <i>ptlldx</i> ]	8	uimsbf
<i>idc_nivel</i> [ <i>ptlldx</i> ].		
}	8	uimsbf
<i>núm_puntos_operativos</i>		
<i>para</i> ( <i>i</i> = 0; <i>i</i> < <i>núm_puntos_operativos</i> ; <i>i</i> ++, <i>opldx</i> ++)	8	uimsbf
{	8	uimsbf
<i>destino_ols</i> [ <i>opldx</i> ]		
<i>esquema_división_destino</i> [ <i>opldx</i> ]	5	uimsbf
<i>reservado</i>	3	bslbf
<i>máx_id_temporal</i> [ <i>opldx</i> ]	2	uimsbf
<i>reservado</i>	6	uimsbf
<i>recuento_es</i> [ <i>opldx</i> ]		
<i>para</i> ( <i>j</i> = 0; <i>j</i> < <i>es_recuento</i> [ <i>opldx</i> ; <i>j</i> ++] {	1	uimsbf
<i>reservado</i>	1	uimsbf
<i>dependencias_antepuestas</i> [ <i>opldx</i> ]	6	uimsbf
<i>referencia_es</i> [ <i>opldx</i> ] [ <i>j</i> ]		
}	6	uimsbf
<i>núm_capas_salida_destino</i> [ <i>opldx</i> ]	4	uimsbf
<i>reservado</i>	6	uimsbf
<i>núm_capas</i> [ <i>opldx</i> ]		
<i>para</i> ( <i>j</i> = 0; <i>j</i> < <i>núm_capas</i> [ <i>opldx</i> ; <i>j</i> ++] {	1	uimsbf
<i>indicador_capa_salida</i> [ <i>opldx</i> ] [ <i>j</i> ]	7	uimsbf
<i>idx_ref_ptl</i> [ <i>opldx</i> ] [ <i>j</i> ]		
}		
<i>reservado</i>	6	uimsbf
<i>idc_información_velocidad_trama</i> [ <i>opldx</i> ]	2	uimsbf
<i>si</i> ( <i>idc_información_velocidad_trama_constante</i> [ <i>opldx</i> ] > 0)		
{	4	bslbf
<i>reservado</i>	12	uimsbf
<i>indicador_velocidad_trama</i> [ <i>opldx</i> ]		

Sintaxis	Nº de bits	Mnemónico
}		
}		
}		

[0113] La semántica ejemplar para los elementos sintácticos añadidos de esta tabla se describe a continuación. La semántica para los otros elementos sintácticos puede quedar igual que en L-HEVC TS.

5 [0114] **núm\_perfil\_grada\_nivel**: campo de 8 bits que especifica el número de estructuras de perfil, grada y nivel señalizadas en este descriptor.

[0115] **espacio\_perfil [ptlldx]**, **indicador\_grada [ptlldx]**, **idc\_perfil [ptlldx]**, **indicación\_compatibilidad\_perfil[ptlldx]**, **indicador\_origen\_progresivo[ptlldx]**, **indicador\_origen\_entrelazado [ptlldx]**, **indicador\_restricción\_no\_empaquetado [ptlldx]**, **indicador\_restricción\_trama\_solamente [ptlldx]**, **reservado\_cero\_44bits [ptlldx]**, **idc\_nivel [ptlldx]** - Estos campos se codificarán de acuerdo a la semántica definida en la Rec. UIT-T H.265 | ISO/CEI 23008-2 para *espacio\_perfil\_general*, *indicador\_grada\_general*, *idc\_perfil\_general*, *indicación\_compatibilidad\_perfil\_general*, *indicador\_origen\_progresivo\_general*, *indicador\_origen\_entrelazado\_general*, *indicador\_restricción\_no\_empaquetado\_general*, *indicador\_restricción\_trama\_solamente\_general*, *reservado\_cero\_44bits\_general*, *idc\_nivel\_general*, respectivamente, para capas de puntos operativos de la HEVC descritos dentro de este descriptor.

10

15

[0116] El valor de pttldx y opldx se inicializan igual a 0 para el primer punto operativo de la HEVC de un programa.

20 [0117] **núm\_puntos\_operativos** - este campo de 8 bits indica el número de puntos operativos descritos en este descriptor.

[0118] **ols\_destino [opldx]** - Un campo de 8 bits que especifica el conjunto de capas de salida asociado al opldx-ésimo punto operativo definido en este descriptor.

25 [0119] **esquema\_división\_destino[opldx]** - Un campo de 8 bits que especifica el esquema de partición del conjunto de capas de salida asociado al opldx-ésimo punto operativo definido en este descriptor.

30 [0120] **máx\_id\_temporal [opldx]** - Un campo de 3 bits que especifica el máximo idTemporal de las unidades de NAL que pertenece al opldx-ésimo punto operativo definido en este descriptor.

[0121] **recuento\_es[opldx]** - Un campo de 8 bits que especifica el número de valores de referencia\_es incluidos en el siguiente grupo de elementos de datos. La agrupación de flujos elementales, de acuerdo a la lista ordenada indicada en el siguiente grupo de elementos de datos, forma un punto operativo de la HEVC. El valor 0xff está reservado.

35 [0122] **dependencias\_antepuestas[opldx][j]**: un indicador de 1 bit que, cuando se fija en 1, indica que el flujo elemental señalado por el elemento sintáctico *índice\_capa\_integrada\_jerarquía* en el descriptor de jerarquía y el descriptor *extensión\_jerarquía*, con el valor del índice de capa de jerarquía especificado por el siguiente elemento sintáctico *referencia\_es [opldx] [j]*, se agregarán a la lista de flujos elementales para el punto operativo de destino antes del flujo elemental señalado por la *referencia\_es[opldx] [j]*. Si el flujo elemental señalado por el *índice\_capa\_integrada\_jerarquía* tiene dependencias adicionales señalizadas por descriptores de jerarquía, estas dependencias se agregarán a la lista de ES de manera recursiva. Cuando no hay ningún descriptor de jerarquía ni descriptores de extensión de jerarquía presentes para cualquiera del flujo elemental referido por el elemento sintáctico *referencia\_ES[opldx][j]*, el valor de *dependencias\_antepuestas[opldx][j]* no será igual a 1.

45 [0123] **referencia\_ES[opldx][j]**: campo de 6 bits que especifica el valor de índice de capa de jerarquía presente en el descriptor de jerarquía que identifica un flujo elemental.

50 [0124] **núm\_capas\_salida\_destino[opldx]**: este campo de 6 bits especifica el valor del número de capas, destinadas a salida para el optldx-ésimo punto operativo definido en este descriptor.

[0125] **núm\_capas[opldx]** - Un campo de 6 bits que especifica el número de capas incluidas en el optldx-ésimo punto operativo definido en este descriptor.

55 [0126] **indicador\_capa\_salida[opldx] [j]** - Un campo de 1 bit que, cuando tiene asignado el valor '1', indica que la j-ésima capa del opldx-ésimo punto operativo definido en este descriptor es una capa de salida. De lo contrario, cuando tiene asignado el valor '0', indica que la j-ésima capa del opldx-ésimo punto operativo definido en este descriptor no es una capa de salida.



**[0127] `idx_ref_ptl[opIdx][j]`** - Un campo de 8 bits que especifica el índice para el perfil, la grada y el nivel que se asigna a la j-ésima capa del opIdx-ésimo punto operativo definido en este descriptor.

**[0128] `idc_información_velocidad_trama_constante[opIdx]`** - Un campo de 2 bits, en combinación con el elemento sintáctico `indicador_velocidad_trama`, como se especifica a continuación, que especifica cómo se determina la velocidad de tramas para el opIdx-ésimo punto operativo asociado definido en este descriptor. El valor de 0 indica que la velocidad de tramas no está especificada para el punto operativo y que el elemento sintáctico `indicador_velocidad_trama` no está presente en este descriptor para el punto operativo.

**[0129] `indicador_velocidad_trama[opIdx]`** - Si `idc_información_velocidad_trama_constante[opIdx]` es igual a 1, este campo de 12 bits indica un número constante de latidos, como se especifica en el descriptor de temporización y HRD de la HEVC, para la distancia en el tiempo entre dos imágenes en el i-ésimo punto operativo definido en este descriptor. De lo contrario, si `idc_información_velocidad_trama_constante[opIdx]` es igual a 2, este campo de 12 bits indica la velocidad de tramas para el punto operativo, medida en tramas por segundo. De lo contrario, si `idc_información_velocidad_trama_constante[opIdx]` es igual a 3, este campo de 12 bits indica la velocidad de tramas para el punto operativo, medida en tramas por 1,001 segundos.

**[0130]** Por lo tanto, un descriptor de acuerdo a la propuesta Tabla Amd7-1 representa un ejemplo de un descriptor que incluye un conjunto de estructuras de perfil, grada y nivel (PTL) y datos que asocian cada una de las capas de cada uno de los puntos operativos a una estructura correspondiente de las estructuras de PTL. Es decir, los elementos enumerados en el ciclo "para ( $i = 0; i < \text{núm\_perfil\_grada\_nivel}; i ++, \text{ptlIdx} ++$ )" representan un ejemplo de un conjunto de estructuras de perfil, grada y nivel (PTL), mientras que los elementos enumerados en el ciclo "para ( $j = 0; j < \text{núm\_capas[opIdx]}; j ++$ )" representan ejemplos de datos que asocian cada una de las capas de cada uno de los puntos operativos con una correspondiente estructura de las estructuras de PTL.

**[0131]** Un descriptor de acuerdo a la propuesta Tabla Amd7-1 también representa un ejemplo de un descriptor que incluye valores para los elementos sintácticos del conjunto de capas de salida de destino, para cada uno de los puntos operativos, en donde los elementos sintácticos del conjunto de capas de salida de destino especifican los conjuntos de capas de salida de destino asociados a los correspondientes puntos operativos. Es decir, los elementos sintácticos `ols_destino[opIdx]` representan ejemplos de valores para los elementos sintácticos del conjunto de capas de salida de destino para cada uno de los puntos operativos, porque estos valores especifican los conjuntos de capas de salida de destino asociados a los puntos operativos correspondientes.

**[0132]** Por consiguiente, el demultiplexor 29 puede extraer el descriptor del flujo de bits para determinar las estructuras de PTL para cada punto operativo del flujo de bits. Es decir, el demultiplexor 29 puede extraer datos dentro de cada elemento de "para ( $i = 0; i < \text{núm\_perfil\_grada\_nivel}; i ++, \text{ptlIdx} ++$ )", tales como los elementos sintácticos `idc_perfil[ptlIdx]`, `indicador_grada[ptlIdx]` e `idc_nivel[ptlIdx]`, para determinar varios conjuntos de información de perfil, grada y nivel. Cada uno de estos conjuntos puede corresponder a una única estructura de datos de PTL, que el demultiplexor 29 puede instanciar, indizados en orden. El demultiplexor 29 puede iterar además por cada capa de cada punto operativo y determinar cuál de las estructuras de datos de PTL corresponde a cada capa de cada punto operativo, en función del elemento sintáctico `idx_ref_ptl[opIdx][j]`. Es decir, "opIdx" representa un índice para un punto operativo, y "j" representa un índice para una capa del punto operativo. Por lo tanto, el valor del elemento sintáctico `idx_ref_ptl[opIdx][j]` representa una de las estructuras de datos de PTL a las que corresponde la j-ésima capa del opIdx-ésimo punto operativo.

**[0133]** Además, el demultiplexor 29 puede determinar un conjunto de capas de salida de destino para cada uno de los puntos operativos. Es decir, para cada uno de los puntos operativos, el demultiplexor 29 puede recuperar un valor para `ols_destino[opIdx]`, que representa el conjunto de capas de salida de destino para el punto operativo representado por "opIdx".

**[0134]** Esta divulgación también describe ejemplos en los que se puede usar un descriptor de jerarquía y un descriptor de extensión de jerarquía para señalar datos para una capa de HEVC (flujo elemental), tal como describir la dependencia (posiblemente, dependencia de capas). Por ejemplo, el descriptor de jerarquía y el descriptor de extensión de jerarquía, de esta divulgación, pueden ser diferentes a los descriptors de jerarquía y a los descriptors de extensión de jerarquía existentes. En consecuencia, cuando esta divulgación utiliza el término "descriptor de extensión de jerarquía" o "descriptor de jerarquía", la divulgación se refiere a versiones actualizadas de dichos descriptors, a menos que de la descripción quede claro que se está haciendo referencia a la versión anterior de dichos descriptors. En algunos ejemplos, cualquiera entre este descriptor de extensión de jerarquía modificada y este descriptor de jerarquía modificados, o ambos, pueden combinarse con el descriptor de punto operativo de la HEVC de la propuesta Tabla Amd7-1, como se ha expuesto anteriormente.

**[0135]** Por consiguiente, el demultiplexor 29 puede configurarse para determinar las dependencias entre flujos elementales usando, por ejemplo, el descriptor de jerarquía y/o el descriptor de extensión de jerarquía, como se ha expuesto anteriormente.

**[0136]** El multiplexor 21 puede configurarse para formar el descriptor de punto operativo de la HEVC, el descriptor

de jerarquía y/o el descriptor de extensión de jerarquía, mientras que el demultiplexor 29 puede usar el descriptor de punto operativo de la HEVC, el descriptor de jerarquía y/o el descriptor de extensión de jerarquía para procesar los datos de vídeo recibidos, por ejemplo, para ensamblar los datos de vídeo en una forma que pueda ser utilizada por el decodificador de vídeo 30. Aunque no se muestra en el ejemplo de la figura 1, un dispositivo intermedio también puede usar estos descriptores, por ejemplo, para realizar una extracción de subflujo de bits. Por ejemplo, un elemento de red consciente de los medios (MANE) puede realizar una extracción del subflujo de bits utilizando el descriptor de punto operativo de la HEVC, el descriptor de jerarquía y/o el descriptor de extensión de jerarquía.

**[0137]** El multiplexor 21, el demultiplexor 29, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden implementarse, cada uno, como cualquiera entre varios circuitos codificadores o decodificadores adecuados, según corresponda, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), formaciones de compuertas programables en el terreno (FPGA), circuitos lógicos discretos, software, hardware, firmware o cualquier combinación de los mismos. Cada uno entre el codificador de vídeo 20 y el decodificador de vídeo 30 puede incluirse en uno o más codificadores o decodificadores, cualquiera de los cuales puede integrarse como parte de un codificador/decodificador de vídeo (CÓDEC) combinado. Un dispositivo que incluye un codificador de vídeo 20 y/o un decodificador de vídeo 30 puede comprender un circuito integrado, un microprocesador y/o un dispositivo de comunicación inalámbrica, tal como un teléfono celular.

**[0138]** De esta manera, el demultiplexor 29 representa un ejemplo de un dispositivo que incluye una memoria para almacenar datos extraídos del flujo de bits, y una o más unidades de procesamiento configuradas para extraer un descriptor del flujo de bits, en donde el flujo de bits incluye capas de datos de vídeo para los puntos operativos, por separado del descriptor, de modo que cada punto operativo incluya una o más de las capas de datos de vídeo, y en el que el descriptor incluya un conjunto de estructuras de perfil, grada y nivel (PTL) y datos que asocian cada una de las capas de cada uno de los puntos operativos con una correspondiente estructura de las estructuras de PTL, extraer datos de vídeo para uno de los puntos operativos del flujo de bits basándose, al menos en parte, en las estructuras de PTL a las que corresponden las capas de dicho uno de los puntos operativos, y proporcionar los datos de vídeo extraídos a un decodificador de vídeo.

**[0139]** 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 a 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 base, donde las capas de mejora pueden corresponder a diferentes dimensiones de ajustabilidad a escala. El codificador de vídeo 20 puede realizar la intracodificación y la intercodificación de bloques de vídeo dentro de fragmentos de vídeo. La intracodificación se basa en la predicción espacial para reducir o eliminar la redundancia espacial en el vídeo dentro de una trama o imagen de vídeo determinada. La intercodificación se basa en la predicción temporal o entre capas para reducir o eliminar la redundancia en el vídeo dentro de tramas o imágenes de una secuencia de vídeo, o de una capa de referencia (por ejemplo, una vista de referencia). La intramodalidad (modalidad I) puede referirse a cualquiera de varias modalidades de codificación de base espacial. Las intermodalidades, tales como la predicción unidireccional (modalidad P) o la bipredicción (modalidad B), pueden referirse a cualquiera de varias modalidades de codificación de base temporal.

**[0140]** Como se muestra en la figura 2, el codificador de vídeo 20 recibe un bloque de vídeo actual dentro de una trama de vídeo a codificar. En el ejemplo de la figura 2, el codificador de vídeo 20 incluye la unidad de selección de modalidad 40, la memoria de imágenes de referencia 64, el sumador 50, la unidad de procesamiento de transformación 52, la unidad de cuantización 54 y la unidad de codificación por entropía 56. La unidad de selección de modalidad 40, a su vez, incluye la unidad de compensación de movimiento 44, la unidad de estimación de movimiento 42, la unidad de intrapredicción 46 y la unidad de partición 48. Para la reconstrucción de bloques de vídeo, el codificador de vídeo 20 también incluye la unidad de cuantización inversa 58, la unidad de transformación inversa 60 y el sumador 62. También se puede incluir un filtro de desbloqueo (no mostrado en la Figura 2) para filtrar las fronteras de los bloques y eliminar las distorsiones de efecto pixelado del vídeo reconstruido. Si se desea, el filtro de desbloqueo normalmente filtraría la salida del sumador 62. También se pueden usar filtros adicionales (en bucle o posteriores al bucle) además del filtro de desbloqueo. Dichos filtros no se muestran por brevedad pero, si se desea, pueden filtrar la salida del sumador 50 (como un filtro en bucle).

**[0141]** Durante el proceso de codificación, el codificador de vídeo 20 recibe un trama o fragmento de vídeo para ser codificado. La trama o el fragmento se pueden dividir en múltiples bloques de vídeo. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 realizan una codificación interpredictiva del bloque de vídeo recibido, en relación con uno o más bloques en una o más tramas de referencia, para proporcionar una predicción temporal. La unidad de intrapredicción 46 puede realizar alternativamente la codificación intrapredictiva del bloque de vídeo recibido, en relación con uno o más bloques vecinos en la misma trama o el mismo fragmento que el bloque a codificar, para proporcionar predicción espacial. El codificador de vídeo 20 puede realizar múltiples pases de codificación, por ejemplo, para seleccionar una modalidad de codificación adecuada para cada bloque de datos de vídeo.

**[0142]** Además, la unidad de partición 48 puede dividir bloques de datos de vídeo en subbloques, basándose en la evaluación de esquemas de partición anteriores en pases de codificación anteriores. Por ejemplo, la unidad de

partición 48 puede dividir inicialmente una trama o fragmento en unas LCU y dividir cada una de las LCU en subCU, en función del análisis de velocidad-distorsión (por ejemplo, optimización de velocidad-distorsión). La unidad de selección de modalidad 40 puede además producir una estructura de datos de árbol cuádruple, indicativa de la partición de una LCU en subCU. Las CU de nodo de hoja del árbol cuádruple pueden incluir una o más PU y una o más TU.

**[0143]** La unidad de selección de modalidad 40 puede seleccionar una de las modalidades de codificación, intracodificación o intercodificación, por ejemplo, en función de resultados erróneos, y proporciona el bloque intrapredicho o interpredicho resultante al sumador 50 para generar datos de bloque residual y al sumador 62 para reconstruir el bloque codificado, para su uso en una trama de referencia. La unidad de selección de modalidad 40 también proporciona elementos sintácticos, tales como vectores de movimiento, indicadores de intramodalidad, información de partición y otra información sintáctica de ese tipo, a la unidad de codificación por entropía 56.

**[0144]** La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden estar sumamente integradas, pero se ilustran por separado con fines conceptuales. La estimación de movimiento, realizada por la unidad de estimación de movimiento 42, es el proceso de generación de vectores de movimiento, que estiman el movimiento para bloques de vídeo. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de una PU de un bloque de vídeo dentro de una trama o imagen de vídeo actual en relación con un bloque predictivo dentro de una trama de referencia (u otra unidad codificada) con respecto al bloque actual que se está codificando dentro de la trama actual (u otra unidad codificada). Un bloque predictivo es un bloque del que se descubre que coincide estrechamente con el bloque que se va a codificar, en términos de diferencia de píxeles, lo que se puede determinar por la suma de la diferencia absoluta (SAD), la suma de la diferencia al cuadrado (SSD) u otras métricas de diferencia. En algunos ejemplos, el codificador de vídeo 20 puede calcular valores para las posiciones fraccionarias de píxeles de imágenes de referencia almacenadas en la memoria de imágenes de referencia 64. Por ejemplo, el codificador de vídeo 20 puede interpolar valores de posiciones de un cuarto de píxel, posiciones de un octavo de píxel u otras posiciones fraccionarias de píxeles de la imagen de referencia. Por lo tanto, la unidad de estimación de movimiento 42 puede realizar una búsqueda de movimiento relativa a las posiciones de píxeles completos y las posiciones fraccionarias de píxeles, y generar un vector de movimiento con una precisión de píxeles fraccionaria.

**[0145]** 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 puede seleccionarse a partir de una primera lista de imágenes de referencia (Lista 0) o una segunda lista de imágenes de referencia (Lista 1), cada una de las cuales identifica una o más imágenes de referencia almacenadas en la memoria de imágenes de referencia 64. La unidad de estimación de movimiento 42 envía el vector de movimiento calculado a la unidad de codificación por entropía 56 y a la unidad de compensación de movimiento 44.

**[0146]** La compensación de movimiento, realizada por la unidad de compensación de movimiento 44, puede implicar capturar o generar el bloque predictivo basándose en el vector de movimiento determinado por la unidad de estimación de movimiento 42. De nuevo, la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden integrarse funcionalmente, en algunos ejemplos. Al recibir el vector de movimiento para la PU del bloque de vídeo actual, la unidad de compensación de movimiento 44 puede ubicar el bloque predictivo al que apunta el vector de movimiento en una de las listas de imágenes de referencia. El sumador 50 forma un bloque de vídeo residual restando los valores de píxeles del bloque predictivo a los valores de píxeles del bloque de vídeo actual que se está codificando, formando valores de diferencia de píxeles, como se expone a continuación. En general, la unidad de estimación de movimiento 42 realiza la estimación de movimiento en relación con los componentes de luma, y la unidad de compensación de movimiento 44 utiliza vectores de movimiento calculados basándose en los componentes de luma, tanto para los componentes de croma como para los componentes de luma. La unidad de selección de modalidad 40 también puede generar elementos sintácticos asociados a los bloques de vídeo y al fragmento de vídeo, para su uso por el decodificador de vídeo 30 en la decodificación de los bloques de vídeo del fragmento de vídeo.

**[0147]** Alternativamente, la unidad de estimación de movimiento 42 puede realizar predicción entre capas (por ejemplo, entre vistas) para un bloque de una imagen en una capa dependiente. Por ejemplo, la unidad de estimación de movimiento 42 puede configurarse para calcular un vector de movimiento de disparidad cuando se realiza la predicción entre vistas de una imagen en una vista dependiente. En otros ejemplos, la unidad de compensación de movimiento 44 puede realizar una predicción de vector de movimiento cero de un bloque al realizar una predicción entre capas, por ejemplo, cuando una capa de mejora corresponde a una dimensión de ajustabilidad a escala, para la cual los bloques en la capa de mejora están situados en la misma, o esencialmente la misma, posición que los bloques en la capa base que se está mejorando. Dichas dimensiones de ajustabilidad a escala pueden incluir, por ejemplo, profundidad de bits de croma, formato de color, gama de colores, PSNR o similares.

**[0148]** La unidad de intrapredicción 46 puede intrapredicir un bloque actual, como alternativa a la interpredicción realizada por la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44, como se ha descrito anteriormente. En particular, la unidad de intrapredicción 46 puede determinar una modalidad de intrapredicción a usar, para codificar un bloque actual. En algunos ejemplos, la unidad de intrapredicción 46 puede codificar un bloque actual usando varias modalidades de intrapredicción, por ejemplo, durante pases de codificación

distintos, y la unidad de intrapredicción 46 (o la unidad de selección de modalidad 40, en algunos ejemplos) puede seleccionar una modalidad de intrapredicción adecuada a usar, a partir de las modalidades probadas.

5 **[0149]** Por ejemplo, la unidad de intrapredicción 46 puede calcular los valores de velocidad-distorsión utilizando un análisis de velocidad-distorsión para las diversas modalidades probadas de intrapredicción, y seleccionar la modalidad de intrapredicción con las mejores características de velocidad-distorsión entre las modalidades probadas. El análisis de velocidad-distorsión generalmente determina una magnitud de distorsión (o error) entre un bloque codificado y un bloque original no codificado, que fue codificado para producir el bloque codificado, así como una velocidad de bits (es decir, una cantidad de bits) utilizada para producir el bloque codificado. La unidad de intrapredicción 46 puede  
10 calcular las razones a partir de las distorsiones y las velocidades para los diversos bloques codificados, para determinar qué modalidad de intrapredicción presenta el mejor valor de velocidad-distorsión para el bloque.

15 **[0150]** Después de seleccionar una modalidad de intrapredicción para un bloque, la unidad de intrapredicción 46 puede proporcionar información, indicativa de la modalidad de intrapredicción seleccionada para el bloque, a la unidad de codificación por entropía 56. La unidad de codificación por entropía 56 puede codificar la información que indica la modalidad de intrapredicción seleccionada. El codificador de vídeo 20 puede incluir en el flujo de bits transmitidos datos de configuración, que pueden incluir una pluralidad de tablas de índices de modalidades de intrapredicción y una pluralidad de tablas modificadas de índices de modalidades de intrapredicción (también denominadas tablas de correlación de palabras de código), definiciones de contextos de codificación para varios bloques e indicaciones de  
20 una modalidad de intrapredicción más probable, una tabla de índices de modalidades de intrapredicción y una tabla modificada de índices de modalidades de intrapredicción, a usar para cada uno de los contextos.

25 **[0151]** El codificador de vídeo 20 forma un bloque de vídeo residual restando los datos de predicción provenientes 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 aplica una transformación, tal como una transformación de coseno discreta (DCT) o una transformación conceptualmente similar, al bloque residual, produciendo un bloque de vídeo que comprende valores de coeficientes de transformación residuales. La unidad de procesamiento de transformaciones 52 puede realizar otras transformaciones que son conceptualmente similares a la DCT. También se podrían usar transformaciones de  
30 ondulaciones, transformaciones enteras, transformaciones de subbanda u otros tipos de transformaciones.

35 **[0152]** En cualquier caso, la unidad de procesamiento de transformación 52 aplica la transformación al bloque residual, produciendo un bloque de coeficientes de transformación residuales. La transformación puede convertir la información residual, desde un dominio de valores de píxel, a un dominio de transformación, tal como un dominio de frecuencia. La unidad de procesamiento de transformación 52 puede enviar los coeficientes de transformación resultantes a la unidad de cuantización 54. La unidad de cuantización 54 cuantiza los coeficientes de transformación para reducir aún más la velocidad de bits. El proceso de cuantización puede reducir la profundidad de bits asociada a algunos de, o todos, los coeficientes. El proceso de cuantización también se puede denominar proceso de "ajuste a escala" y, por lo tanto, los coeficientes de transformación cuantizados también se pueden denominar "coeficientes de  
40 transformación ajustados a escala". El grado de cuantización (o de ajuste a escala) puede modificarse ajustando un parámetro de cuantización. En algunos ejemplos, la unidad de codificación por entropía 56 puede entonces realizar un recorrido de la matriz que incluye los coeficientes de transformación cuantizados.

45 **[0153]** Después de la cuantización, la unidad de codificación por entropía 56 codifica por entropía los coeficientes de transformación cuantizados y recorridos. Por ejemplo, la unidad de codificación por entropía 56 puede realizar codificación de longitud variable adaptativa de contexto (CAVLC), codificación de aritmética binaria adaptativa de contexto (CABAC), codificación de aritmética binaria adaptativa de contexto (SBAC) basada en la sintaxis, codificación por entropía y división en intervalos de probabilidad, u otra técnica de codificación por entropía. En el caso de la codificación por entropía basada en el contexto, el contexto puede basarse en bloques vecinos. A continuación de la  
50 codificación por entropía mediante 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 archivar para su posterior transmisión o recuperación.

55 **[0154]** La unidad de cuantización inversa 58 y la unidad de transformación inversa 60 aplican la cuantización inversa y la transformación inversa, respectivamente, para reconstruir el bloque residual en el dominio de píxeles, por ejemplo, para su uso posterior como un bloque de referencia. La unidad de compensación de movimiento 44 puede calcular un bloque de referencia sumando el bloque residual a un bloque predictivo de una de las tramas de la memoria de imágenes de referencia 64. La unidad de compensación de movimiento 44 también puede aplicar uno o más filtros de interpolación al bloque residual reconstruido para calcular los valores fraccionarios de píxel, para su uso en la  
60 estimación de movimiento. El sumador 62 suma el bloque residual reconstruido al bloque de predicción compensado por movimiento, producido por la unidad de compensación de movimiento 44, para producir un bloque de vídeo reconstruido para su almacenamiento en la memoria de imágenes de referencia 64. El bloque de vídeo reconstruido puede ser usado por la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 como un bloque de referencia para intercodificar un bloque en un trama de vídeo posterior.

65 **[0155]** La figura 3 es un diagrama de bloques que ilustra un ejemplo del decodificador de vídeo 30 que puede

implementar técnicas para transportar datos de vídeo codificados de acuerdo a las extensiones de una norma de codificación de vídeo. En el ejemplo de la figura 3, el decodificador de vídeo 30 incluye una unidad de decodificación por entropía 70, una unidad de compensación de movimiento 72, una unidad de intrapredicción 74, una unidad de cuantización inversa 76, una unidad de transformación inversa 78, una memoria de imágenes de referencia 82 y un sumador 80. El decodificador de vídeo 30 puede, en algunos ejemplos, realizar un pase de decodificación generalmente recíproco al pase de codificación descrito con respecto al codificador de vídeo 20 (figura 2). La unidad de compensación de movimiento 72 puede generar datos de predicción basándose en vectores de movimiento recibidos desde la unidad de decodificación por entropía 70, mientras que la unidad de intrapredicción 74 puede generar datos de predicción basándose en indicadores de modalidad de intrapredicción, recibidos desde la unidad de decodificación por entropía 70.

**[0156]** Durante el proceso de decodificación, el decodificador de vídeo 30 recibe un flujo de bits de vídeo codificado que representa bloques de vídeo de un fragmento de vídeo codificado y elementos sintácticos asociados, desde el codificador de vídeo 20. La unidad de decodificación por entropía 70 del decodificador de vídeo 30 decodifica por entropía el flujo de bits para generar coeficientes cuantizados, vectores de movimiento o indicadores de modalidad de intrapredicción y otros elementos sintácticos. La unidad de decodificación por entropía 70 remite los vectores de movimiento y otros elementos sintácticos a la unidad de compensación de movimiento 72. El decodificador de vídeo 30 puede recibir los elementos sintácticos en el nivel de fragmento de vídeo y/o el nivel de bloque de vídeo.

**[0157]** Cuando el fragmento de vídeo se codifica como un fragmento intracodificado (I), la unidad de intrapredicción 74 puede generar datos de predicción para un bloque de vídeo del fragmento de vídeo actual, basándose en una modalidad de intrapredicción señalizada y en datos de bloques decodificados previamente de la trama o imagen actual. Cuando la trama de vídeo se codifica como un fragmento intercodificado (es decir, B o P), 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 pueden producirse a partir de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. El decodificador de vídeo 30 puede construir las listas de tramas de referencia, Lista 0 y Lista 1, usando técnicas de construcción predeterminadas basadas en imágenes de referencia almacenadas en la memoria de imágenes de referencia 82.

**[0158]** La unidad de compensación de movimiento 72 determina la información de predicción para un bloque de vídeo del fragmento de vídeo actual, analizando sintácticamente los vectores de movimiento y otros elementos sintácticos, y utiliza la información de predicción para producir los bloques predictivos para el bloque de vídeo actual que se está decodificando. Por ejemplo, la unidad de compensación de movimiento 72 utiliza algunos de los elementos sintácticos recibidos para determinar una modalidad de predicción (por ejemplo, intrapredicción o interpredicción) que se utiliza para codificar los bloques de vídeo del fragmento de vídeo, un tipo de fragmento de interpredicción (por ejemplo, fragmento B o fragmento P), información de construcción para una o más de las listas de imágenes de referencia para el fragmento, vectores de movimiento para cada bloque de vídeo intercodificado del fragmento, estado de interpredicción para cada bloque de vídeo intercodificado del fragmento y otra información, para decodificar los bloques de vídeo en el fragmento de vídeo actual.

**[0159]** La unidad de compensación de movimiento 72 también puede realizar una interpolación basándose en filtros de interpolación. La unidad de compensación de movimiento 72 puede usar filtros de interpolación como los que utiliza el codificador de vídeo 20 durante la codificación de los bloques de vídeo para calcular los valores interpolados para los píxeles fraccionarios de los bloques de referencia. En este caso, la unidad de compensación de movimiento 72 puede determinar los filtros de interpolación utilizados por el codificador de vídeo 20 a partir de los elementos sintácticos recibidos y utilizar los filtros de interpolación para producir bloques predictivos.

**[0160]** En algunos ejemplos, la unidad de compensación de movimiento 72 puede realizar una predicción de vector de movimiento cero de un bloque al realizar una predicción entre capas, por ejemplo, cuando una capa de mejora corresponde a una dimensión de ajustabilidad a escala, para la cual los bloques en la capa de mejora se sitúan en la misma, o esencialmente la misma, posición que los bloques en la capa base que se está mejorando. Dichas dimensiones de ajustabilidad a escala pueden incluir, por ejemplo, profundidad de bits de croma, formato de color, gama de colores, PSNR o similares. Alternativamente, la unidad de compensación de movimiento 72 puede usar vectores de movimiento de disparidad para predecir bloques de una vista dependiente a partir de una o más vistas de referencia (por ejemplo, una vista base). Debería entenderse que una vista es un ejemplo de una capa. Es decir, cuando una capa de mejora es una vista, la dimensión de ajustabilidad a escala puede corresponder a una dimensión de vista (por ejemplo, para proporcionar datos para producir un efecto tridimensional para un espectador).

**[0161]** La unidad de cuantización inversa 76 cuantiza de manera inversa, es decir, decuantiza, los coeficientes de transformación cuantizados proporcionados en el flujo de bits y decodificados por la unidad de decodificación por entropía 70. El proceso de cuantización inversa puede incluir el uso de un parámetro de cuantización QPY calculado por el decodificador de vídeo 30 para cada bloque de vídeo en el fragmento de vídeo, para determinar un grado de cuantización y, de igual modo, un grado de cuantización inversa que debería aplicarse. La unidad de transformación inversa 78 aplica una transformación inversa, por ejemplo, una DCT inversa, una transformación entera inversa o un proceso de transformación inversa conceptualmente similar, a los coeficientes de transformación con el fin de producir

bloques residuales en el dominio de píxeles.

**[0162]** 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 forma un bloque de vídeo decodificado sumando los bloques residuales de la unidad de transformación inversa 78 con 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. Si se desea, también se puede aplicar un filtro de desbloqueo para filtrar los bloques decodificados con el fin de eliminar las distorsiones de efecto pixelado. También se pueden usar otros filtros de bucle (ya sea en el bucle de codificación o después del bucle de codificación) para allanar las transiciones de píxeles, o mejorar la calidad del vídeo. Los bloques de vídeo decodificados en una trama o imagen dadas se almacenan luego en la memoria de imágenes de referencia 82, que almacena las imágenes de referencia utilizadas para la compensación de movimiento posterior. La memoria de imágenes de referencia 82 también almacena vídeo decodificado para una presentación posterior en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la figura 1.

**[0163]** La figura 4 es un diagrama de bloques que ilustra un sistema ejemplar 100 en el que el dispositivo de origen de audio/vídeo (A/V) 120 transporta datos de audio y vídeo al dispositivo de destino de A/V 140. El sistema 100 de la figura 4 puede corresponder a un sistema de videoconferencia, un sistema de servidor/cliente, un sistema difusor/receptor o cualquier otro sistema en el que los datos de vídeo se envíen desde un dispositivo de origen, tal como el dispositivo de origen de A/V 120, a un dispositivo de destino, tal como el dispositivo de destino de A/V 140. En algunos ejemplos, el dispositivo de origen de A/V 120 y el dispositivo de destino de A/V 140 pueden realizar intercambio de información bidireccional. Es decir, el dispositivo de origen de A/V 120 y el dispositivo de destino de A/V 140 pueden ser capaces tanto de codificar como de decodificar (y transmitir y recibir) datos de audio y vídeo. En algunos ejemplos, el codificador de audio 126 puede comprender un codificador de voz, también denominado vocoder.

**[0164]** El dispositivo de origen de A/V 120, en el ejemplo de la figura 4, comprende el origen de audio 122 y el origen de vídeo 124. El origen de audio 122 puede comprender, por ejemplo, un micrófono que produce señales eléctricas representativas de los datos de audio capturados para ser codificados por el codificador de audio 126. Alternativamente, el origen de audio 122 puede comprender un medio de almacenamiento que almacena datos de audio previamente grabados, un generador de datos de audio tal como un sintetizador informatizado o cualquier otro origen de datos de audio. El origen de vídeo 124 puede comprender una cámara de vídeo que produce datos de vídeo para ser codificados por el codificador de vídeo 128, un medio de almacenamiento codificado con datos de vídeo grabados previamente, una unidad de generación de datos de vídeo o cualquier otro origen de datos de vídeo.

**[0165]** Los datos de audio y vídeo "en bruto" (es decir, datos capturados o adquiridos que no han sido codificados) pueden comprender datos analógicos o digitales. Los datos analógicos pueden digitalizarse antes de ser codificados por el codificador de audio 126 y/o el codificador de vídeo 128. El origen de audio 122 puede obtener datos de audio de un orador participante, mientras el orador participante está hablando, y el origen de vídeo 124 puede obtener simultáneamente datos de vídeo del orador participante. En otros ejemplos, el origen de audio 122 puede comprender un medio de almacenamiento legible por ordenador que comprende datos de audio almacenados, y el origen de vídeo 124 puede comprender un medio de almacenamiento legible por ordenador que comprende datos de vídeo almacenados. De esta manera, las técnicas descritas en esta divulgación se pueden aplicar a datos de audio y vídeo en vivo, en transmisión continua, en tiempo real o a datos de audio y vídeo pregrabados y archivados.

**[0166]** Las tramas de audio que corresponden a tramas de vídeo son generalmente tramas de audio que contienen datos de audio que fueron capturados por el origen de audio 122 al mismo tiempo que los datos de vídeo capturados por el origen de vídeo 124 que están contenidos dentro de las tramas de vídeo. Por ejemplo, mientras un orador participante generalmente produce datos de audio al hablar, el origen de audio 122 captura los datos de audio, y el origen de vídeo 124 captura los datos de vídeo del orador participante al mismo tiempo, es decir, mientras el origen de audio 122 está capturando los datos de audio. Por lo tanto, una trama de audio puede corresponder temporalmente a una o más tramas de vídeo en particular. Por consiguiente, una trama de audio correspondiente a una trama de vídeo generalmente corresponde a una situación en la que los datos de audio y los datos de vídeo se capturaron al mismo tiempo y para los cuales un trama de audio y un trama de vídeo comprenden, respectivamente, los datos de audio y los datos de vídeo que fueron capturados al mismo tiempo.

**[0167]** En algunos ejemplos, el codificador de audio 126 puede codificar un sello cronológico en cada trama de audio codificado que representa un momento en el que se grabaron los datos de audio para la trama de audio codificado y, de manera similar, el codificador de vídeo 128 puede codificar un sello cronológico en cada trama de vídeo codificado que representa un momento en el que se grabaron los datos de vídeo para la trama de vídeo codificado. En tales ejemplos, una trama de audio correspondiente a una trama de vídeo puede comprender una trama de audio que comprende un sello cronológico y una trama de vídeo que comprende el mismo sello cronológico. El dispositivo de origen de audio y vídeo 120 puede incluir un reloj interno desde el cual el codificador de audio 126 y/o el codificador de vídeo 128 pueden generar los sellos cronológicos, o que el origen de audio 122 y el origen de vídeo 124 pueden usar para asociar datos de audio y vídeo, respectivamente, con un sello cronológico.

**[0168]** En algunos ejemplos, el origen de audio 122 puede enviar datos al codificador de audio 126, correspondientes

a una hora en la que se grabaron datos de audio, y el origen de vídeo 124 puede enviar datos al codificador de vídeo 128, correspondientes a una hora en la que se grabaron los datos de vídeo. En algunos ejemplos, el codificador de audio 126 puede codificar un identificador de secuencia en datos de audio codificados para indicar un ordenamiento temporal relativo de datos de audio codificados, pero sin indicar necesariamente una hora absoluta en la que se grabaron los datos de audio y, de manera similar, el codificador de vídeo 128 también puede usar identificadores de secuencia para indicar un ordenamiento temporal relativo de datos de vídeo codificados. De manera similar, en algunos ejemplos, un identificador de secuencia puede asociarse o correlacionarse de otro modo con un sello cronológico.

**[0169]** Las técnicas de esta divulgación están generalmente dirigidas al transporte de datos de multimedios codificados (por ejemplo, audio y/o vídeo), y a la recepción y posterior interpretación y decodificación de los datos de multimedios transportados. Las técnicas de esta divulgación son particularmente aplicables al transporte de datos de codificación de vídeo de vista múltiple (MVC), es decir, datos de vídeo que comprenden una pluralidad de vistas. Como se muestra en el ejemplo de la figura 4, el origen de vídeo 124 puede proporcionar una pluralidad de vistas de una escena al codificador de vídeo 128. La codificación de vista múltiple puede ser útil para generar datos de vídeo tridimensionales, para ser utilizados por una pantalla tridimensional, tal como una pantalla tridimensional estereoscópica o autoestereoscópica.

**[0170]** El dispositivo de origen de A/V 120 puede proporcionar un "servicio" al dispositivo de destino de A/V 140. Un servicio generalmente corresponde a un subconjunto de vistas disponibles de datos de vista múltiple. Por ejemplo, los datos de vista múltiple pueden estar disponibles para ocho vistas, ordenadas de cero a siete. Un servicio puede corresponder a un vídeo estéreo que tiene dos vistas, mientras que otro servicio puede corresponder a cuatro vistas, y otro servicio más puede corresponder a las ocho vistas. En general, un servicio corresponde a cualquier combinación (es decir, cualquier subconjunto) de las vistas disponibles. Un servicio también puede corresponder a una combinación de vistas disponibles y datos de audio. Un punto operativo puede corresponder a un servicio, de modo que el dispositivo de origen de A/V 120 pueda proporcionar además un descriptor de punto operativo para cada servicio proporcionado por el dispositivo de origen de A/V 120.

**[0171]** El dispositivo de origen de A/V 120, de acuerdo a las técnicas de esta divulgación, puede proporcionar servicios que corresponden a un subconjunto de vistas. En general, una vista está representada por un identificador de vista, también denominado "id\_vista". Los identificadores de vista generalmente comprenden elementos sintácticos que pueden usarse para identificar una vista. Un codificador de la MVC proporciona el identificador de vista de una vista cuando la vista está codificada. El identificador de la vista puede ser utilizado por un decodificador de la MVC para la predicción entre vistas, o por otras unidades para otros fines, por ejemplo, para la representación.

**[0172]** La predicción entre vistas es una técnica para codificar datos de vídeo de la MVC de una trama con referencia a una o más tramas en una ubicación temporal común como la trama codificada de diferentes vistas. En general, una trama codificada de datos de vídeo de la MVC puede codificarse de manera predictiva, espacialmente, temporalmente y/o con referencia a tramas de otras vistas en una ubicación temporal común. Por consiguiente, las vistas de referencia, a partir de las cuales se predicen otras vistas, generalmente se decodifican antes de las vistas para las cuales las vistas de referencia actúan como referencia, de modo que estas vistas decodificadas se puedan usar como referencia cuando se decodifican vistas referenciales. El orden de decodificación no se corresponde necesariamente con el orden de los id\_vista. Por lo tanto, el orden de decodificación de las vistas se describe utilizando índices de orden de vista. Los índices de orden de vista son índices que indican el orden de decodificación de los componentes de vista correspondientes en una unidad de acceso.

**[0173]** Cada flujo de datos individual (ya sea audio o vídeo) se conoce como flujo elemental. Un flujo elemental es un componente único, codificado digitalmente (posiblemente comprimido) de un programa. Por ejemplo, la parte de vídeo codificado o la parte de audio codificado del programa puede ser un flujo elemental. Un flujo elemental se puede convertir 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 Identificador de flujo para distinguir los paquetes de PES que pertenecen a un flujo elemental o a otro. La unidad básica de datos de un flujo elemental es un paquete de flujo elemental empaquetado (PES). Por lo tanto, cada vista de datos de vídeo de la MVC corresponde a respectivos flujos elementales. De manera similar, los datos de audio corresponden a uno o más respectivos flujos elementales.

**[0174]** Una secuencia de vídeo codificada de la MVC se puede separar en varios subflujos de bits, cada uno de los cuales es un flujo elemental. Cada subflujo de bits puede identificarse utilizando un subconjunto de los id\_vista de la MVC. Basándose en el concepto de cada subconjunto de los id\_vista de la MVC, se define un subflujo de bits de vídeo de la MVC. Un subflujo de bits de vídeo de la MVC contiene las unidades de NAL de las vistas enumeradas en el subconjunto de los id\_vista de la MVC. Un flujo de programa generalmente contiene solo las unidades de NAL que son las de los flujos elementales. También está diseñado que dos flujos elementales no puedan contener una vista idéntica.

**[0175]** En el ejemplo de la figura 4, el multiplexor 130 recibe flujos elementales que comprenden datos de vídeo del codificador de vídeo 128 y flujos elementales que comprenden datos de audio del codificador de audio 126. En algunos ejemplos, el codificador de vídeo 128 y el codificador de audio 126 pueden incluir, cada uno, empaquetadores para

formar paquetes de PES a partir de datos codificados. En otros ejemplos, el codificador de vídeo 128 y el codificador de audio 126 pueden interconectarse con empaquetadores respectivos para formar paquetes de PES a partir de datos codificados. En aún otros ejemplos, el multiplexor 130 puede incluir empaquetadores para formar paquetes de PES a partir de datos de audio y vídeo codificados.

**[0176]** Un "programa", como se usa en esta divulgación, puede comprender una combinación de datos de audio y datos de vídeo, por ejemplo, un flujo elemental de audio y un subconjunto de vistas disponibles entregadas por un servicio del dispositivo de origen de A/V 120. Cada paquete de PES incluye un `id_flujo` que identifica el flujo elemental al que pertenece el paquete de PES. El multiplexor 130 es responsable de ensamblar flujos elementales en flujos de programa constituyentes o flujos de transporte. Un flujo de programa y un flujo de transporte son dos multiplex alternativos que se orientan a diferentes aplicaciones.

**[0177]** En general, un flujo de programa incluye datos para un programa, mientras que un flujo de transporte puede incluir datos para uno o más programas. El multiplexor 130 puede codificar cualquiera entre un flujo de programa o un flujo de transporte, o ambos, en función de un servicio que se proporciona, un medio al que se pasará el flujo, una serie de programas que se enviarán u otras consideraciones. Por ejemplo, cuando los datos de vídeo se han de codificar en un medio de almacenamiento, puede ser más probable que el multiplexor 130 forme un flujo de programa, mientras que cuando los datos de vídeo han de ser transmitidos continuamente a través de una red, difundidos o enviados como parte de la videotelefonía, es más probable que el multiplexor 130 use un flujo de transporte.

**[0178]** El multiplexor 130 puede estar inclinado a favor de usar un flujo de programa para el almacenamiento y visualización de un solo programa desde un servicio de almacenamiento digital. Un flujo de programa está concebido para su uso en entornos sin errores o en entornos menos susceptibles a encontrar errores, ya que los flujos de programas son bastante susceptibles a errores. Un flujo de programa simplemente comprende los flujos elementales que le pertenecen y generalmente contiene paquetes de longitudes variables. En un flujo de programa, los paquetes de PES que se obtienen de los flujos elementales contribuyentes se organizan en "cajetillas". Una cajetilla comprende un encabezado de cajetilla, un encabezado de sistema optativo y cualquier número de paquetes de 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 velocidad máxima de datos, la cantidad de flujos elementales de audio y vídeo que contribuyen, información adicional de temporización u otra información. Un decodificador puede usar la información contenida en un encabezado del sistema para determinar si el decodificador es capaz o no de decodificar el flujo del programa.

**[0179]** El multiplexor 130 puede usar un flujo de transporte para la entrega simultánea de una pluralidad de programas por canales potencialmente propensos a errores. Un flujo de transporte es un multiplex ideado para aplicaciones de múltiples programas, tales como la difusión, de modo que un solo flujo de transporte pueda alojar muchos programas independientes. Un flujo de transporte puede comprender una sucesión de paquetes de transporte, teniendo cada uno de los paquetes de transporte una longitud de 188 octetos. El uso de paquetes cortos de longitud fija hace que el flujo de transporte sea menos susceptible a errores que el flujo del programa. Además, a cada paquete de transporte de 188 octetos de largo se le puede dar protección adicional contra errores procesando el paquete a través de un proceso estándar de protección contra errores, tal como la codificación de Reed-Solomon. La mejora de la capacidad de recuperación ante errores del flujo de transporte significa que tiene una mejor posibilidad de sobrevivir a los canales propensos a errores que se encuentran en un entorno de difusión, por ejemplo.

**[0180]** Podría parecer que el flujo de transporte es mejor que un flujo de programa debido a su acrecentada capacidad de recuperación ante errores y su capacidad para transportar muchos programas simultáneos. Sin embargo, el flujo de transporte es un multiplex más sofisticado que el flujo de programa y, por consiguiente, es más difícil de crear y más complicado de demultiplexar que un flujo de programa. El primer octeto de un paquete de transporte puede ser un octeto de sincronización con un valor de 0x47 (hexadecimal 47, binario '01000111,' decimal 71). Un único flujo de transporte puede llevar muchos programas diferentes, comprendiendo cada programa muchos flujos elementales empaquetados. El multiplexor 130 puede usar un campo de identificador de paquete (PID) de trece bits para distinguir los paquetes de transporte que contienen los datos de un flujo elemental de los que llevan los datos de otros flujos elementales. Es responsabilidad del multiplexor garantizar que a cada flujo elemental se le conceda un valor de PID único. El último octeto de un paquete de transporte puede ser el campo de recuento de continuidad. El multiplexor 130 incrementa el valor del campo de recuento de continuidad entre paquetes de transporte sucesivos que pertenecen al mismo flujo elemental. Esto permite que un decodificador u otra unidad de un dispositivo de destino, tal como el dispositivo de destino de A/V 140, detecte la pérdida o ganancia de un paquete de transporte y, con un poco de suerte, oculte los errores que de lo contrario podrían resultar de tal suceso.

**[0181]** El multiplexor 130 recibe paquetes de PES para flujos elementales de un programa desde el codificador de audio 126 y el codificador de vídeo 128, y forma las correspondientes unidades de capa de abstracción de red (NAL) de los paquetes de PES. En el ejemplo de la H.264/AVC (codificación avanzada de vídeo), los fragmentos de vídeo codificados se organizan en unidades de NAL, que brindan una representación de vídeo "amigable con la red" que se orienta a aplicaciones tales como la videotelefonía, el almacenamiento, la difusión o la transmisión continua. Las unidades de NAL se pueden categorizar como unidades de NAL de capa de codificación de vídeo (VCL) y unidades de NAL no de VCL. Las unidades de VCL contienen el motor de compresión central y pueden comprender niveles de



bloque, macrobloque y/o fragmento. Otras unidades de NAL son unidades de NAL no de VCL.

**[0182]** El multiplexor 130 puede formar unidades de NAL que comprenden un encabezado que identifica un programa al que pertenece la NAL, así como una carga útil, por ejemplo, datos de audio, datos de vídeo o datos que describen el flujo de transporte o de programa al que corresponde la unidad de NAL. Por ejemplo, en la norma H.264/AVC, una unidad de NAL incluye un encabezado de 1 octeto y una carga útil de tamaño variable. En un ejemplo, un encabezado de unidad de NAL comprende un elemento `id_prioridad`, un elemento `id_temporal`, un elemento `indicador_img_ancla`, un elemento `id_vista`, un elemento `indicador_no_idr` y un elemento `indicador_entre_vistas`. En la MVC convencional, la unidad de NAL definida por la norma H.264 se conserva, a excepción de las unidades de NAL con prefijo y las unidades de NAL de fragmento codificado según la MVC, que incluyen un encabezado de unidad de NAL de la MVC de 4 octetos y la carga útil de la unidad de NAL.

**[0183]** El elemento de identificador de prioridad de un encabezado de NAL puede usarse para un proceso sencillo de adaptación de flujo de bits de un solo trayecto. El elemento `id_temporal` se puede usar para especificar el nivel temporal de la unidad de NAL correspondiente, donde los diferentes niveles temporales corresponden a diferentes velocidades de trama.

**[0184]** El elemento `indicador_img_ancla` puede indicar si una imagen es una imagen de anclaje o una imagen no de anclaje. Las imágenes de anclaje y todas las imágenes sucesivas en el orden de salida (es decir, el orden de visualización) se pueden decodificar correctamente sin decodificar las imágenes anteriores en el orden de decodificación (es decir, el orden del flujo de bits) y, por lo tanto, se pueden usar como puntos de acceso aleatorios. Las imágenes de anclaje y las imágenes no de anclaje pueden tener diferentes dependencias, las cuales se señalizan en el conjunto de parámetros de secuencia. Otros indicadores se han de exponer y usar en las siguientes secciones de este capítulo. Una imagen de anclaje de ese tipo también puede denominarse un punto de acceso abierto de GOP (Grupo de imágenes), mientras que un punto de acceso cerrado de GOP también dispone de soporte cuando el elemento `indicador_no_idr` es igual a cero. El elemento `indicador_no_idr` indica si una imagen es una imagen de actualización instantánea del decodificador (IDR) o de IDR de vista (V-IDR). En general, una imagen de IDR, y todas las imágenes que la suceden en orden de salida o en orden de flujo de bits, pueden decodificarse correctamente sin decodificar las imágenes anteriores, ya sea en orden de decodificación o en orden de visualización.

**[0185]** El elemento identificador de vista puede comprender información sintáctica que se puede usar para identificar una vista, que se puede usar para la interactividad de datos dentro de un decodificador de la MVC, por ejemplo, para predicción entre vistas, y fuera de un decodificador, por ejemplo, para la representación. El elemento `indicador_entre_vistas` puede especificar si otras vistas usan la unidad de NAL correspondiente para la predicción entre vistas. Para transmitir la información del encabezado de la unidad de NAL de 4 octetos para una vista base, que puede ser conforme a la AVC, se define una unidad de NAL de prefijo en la MVC. En el contexto de la MVC, la unidad de acceso a la vista base incluye las unidades de NAL de VCL de la instancia temporal actual de la vista, así como su unidad de NAL con prefijo, que contiene solo la cabecera de la unidad de NAL. Un decodificador de la norma H.264/AVC puede ignorar la unidad de NAL con prefijo.

**[0186]** Una unidad de NAL que incluye datos de vídeo en su carga útil puede comprender varios niveles de granularidad de datos de vídeo. Por ejemplo, una unidad de NAL puede comprender un bloque de datos de vídeo, un macrobloque, una pluralidad de macrobloques, un fragmento de datos de vídeo o una trama completa de datos de vídeo. El multiplexor 130 puede recibir datos de vídeo codificados desde el codificador de vídeo 128, en forma de paquetes de PES de flujos elementales. El multiplexor 130 puede asociar cada flujo elemental con un programa correspondiente correlacionando los `id_flujo` con los correspondientes programas, por ejemplo, en una base de datos u otra estructura de datos, tal como una Tabla de mapas de programa (PMT) o un Mapa de flujos de programa (PSM).

**[0187]** El multiplexor 130 también puede ensamblar unidades de acceso a partir de una pluralidad de unidades de NAL. En general, una unidad de acceso puede comprender una o más unidades de NAL para representar una trama de datos de vídeo, así como datos de audio correspondientes a la trama cuando dichos datos de audio están disponibles. Una unidad de acceso generalmente incluye todas las unidades de NAL para una instancia temporal de salida, por ejemplo, todos los datos de audio y vídeo para una instancia temporal. Por ejemplo, si cada vista tiene una velocidad de tramas de 120 tramas por segundo (fps), entonces cada instancia temporal puede corresponder a un intervalo de tiempo de 0,05 segundos. Durante este intervalo de tiempo, las tramas específicas para todas las vistas de la misma unidad de acceso (la misma instancia temporal) pueden representarse simultáneamente. En un ejemplo correspondiente a la norma H.264/AVC, una unidad de acceso puede comprender una imagen codificada en una instancia temporal, que puede presentarse como una imagen codificada primaria. Por consiguiente, una unidad de acceso puede comprender todas las tramas de audio y vídeo de una instancia temporal común, por ejemplo, todas las vistas correspondientes al momento X. Esta divulgación también se refiere a una imagen codificada de una vista particular como un "componente de vista". Es decir, un componente de vista puede comprender una imagen (o trama) codificada para una vista particular en un momento determinado. Por consiguiente, una unidad de acceso puede definirse como que comprende todos los componentes de vista de una instancia temporal común. El orden de decodificación de las unidades de acceso no tiene que ser necesariamente el mismo que el orden de salida o visualización.

**[0188]** El multiplexor 130 también puede incrustar datos relativos a un programa en una unidad de NAL. Por ejemplo, el multiplexor 130 puede crear una unidad de NAL que comprende una Tabla de mapas de programas (PMT) o un Mapa de flujos de programas (PSM). En general, una PMT se usa para describir un flujo de transporte, mientras que un PSM se usa para describir un flujo de programa. Como se describe con mayor detalle con respecto al ejemplo de la figura 2 a continuación, el multiplexor 130 puede comprender o interactuar con una unidad de almacenamiento de datos que asocia flujos elementales recibidos desde el codificador de audio 126 y el codificador de vídeo 128 con programas y, en consecuencia, con respectivos flujos de transporte y/o flujos de programas.

**[0189]** Al igual que con la mayoría de las normas de codificación de vídeo, H.264/AVC y HEVC definen la sintaxis, la semántica y el proceso de decodificación para flujos de bits sin errores, cualquiera de los cuales es conforme a un determinado perfil o nivel. Estas normas no especifican el codificador, pero el codificador tiene la tarea de garantizar que los flujos de bits generados sean conformes a las normas para un decodificador. En el contexto de una norma de codificación de vídeo, un "perfil" corresponde a un subconjunto de algoritmos, características o herramientas y restricciones que se aplican a ellos. Según lo definido por la norma H.264, por ejemplo, un "perfil" es un subconjunto de la sintaxis completa del flujo de bits, que es especificada por la norma H.264. Un "nivel" corresponde a las limitaciones del consumo de recursos del decodificador, tales como, por ejemplo, la memoria y el cálculo del decodificador, que están relacionados con la resolución de las imágenes, la velocidad de bits y la velocidad de procesamiento de macrobloques (MB).

**[0190]** La norma H.264, por ejemplo, reconoce que, dentro de los límites impuestos por la sintaxis de un perfil dado, todavía es posible requerir una gran variación en el rendimiento de los codificadores y decodificadores, según los valores tomados por los elementos sintácticos en el flujo de bits, tales como el tamaño especificado de las imágenes decodificadas. La norma H.264 reconoce además que, en muchas aplicaciones, no es práctico ni económico implementar un decodificador capaz de tratar todos los usos hipotéticos de la sintaxis dentro de un perfil particular. En consecuencia, la norma H.264 define un "nivel" como un conjunto especificado de restricciones impuestas sobre los valores de los elementos sintácticos en el flujo de bits. Estas restricciones pueden ser simples límites sobre los valores. Alternativamente, estas restricciones pueden adoptar la forma de restricciones sobre combinaciones aritméticas de valores (por ejemplo, el ancho de la imagen multiplicado por la altura de la imagen multiplicado por el número de imágenes decodificadas por segundo). La norma H.264 además establece que las implementaciones individuales pueden prestar soporte a un nivel diferente para cada perfil con soporte.

**[0191]** Un decodificador conforme a un perfil habitualmente presta soporte a todas las características definidas en el perfil. Por ejemplo, como característica de la codificación, la codificación de imágenes B no tiene soporte en el perfil de línea de base de la norma H.264/AVC, pero sí en otros perfiles de la norma H.264/AVC. Un decodificador conforme a un nivel debería ser capaz de decodificar cualquier flujo de bits que no requiera recursos más allá de las limitaciones definidas en el nivel. Las definiciones de perfiles y niveles pueden ser útiles para la interpretabilidad. Por ejemplo, durante la transmisión de vídeo, se pueden negociar y acordar un par de definiciones de perfil y nivel para una sesión de transmisión completa. Más específicamente, en la norma H.264/AVC, un nivel puede definir, por ejemplo, limitaciones en el número de macrobloques que deben procesarse, el tamaño del almacén temporal de imágenes decodificadas (DPB), el tamaño del almacén temporal de imágenes codificadas (CPB), el rango vertical del vector de movimiento, el número máximo de vectores de movimiento por dos MB consecutivos, y si un bloque B puede tener particiones de submacrobloques de menos de 8x8 píxeles. De esta manera, un decodificador puede determinar si el decodificador es capaz de decodificar adecuadamente el flujo de bits.

**[0192]** Los conjuntos de parámetros generalmente contienen información de encabezado de capa de secuencia en conjuntos de parámetros de secuencia (SPS) y la información, de cambios infrecuentes, de encabezado de capa de imagen en conjuntos de parámetros de imagen (PPS). Con los conjuntos de parámetros, esta información que cambia con poca frecuencia no necesita repetirse para cada secuencia o imagen; por lo tanto, la eficacia de codificación puede mejorarse. Además, el uso de conjuntos de parámetros puede permitir la transmisión fuera de banda de información de encabezado, evitando la necesidad de transmisiones redundantes para lograr la capacidad de recuperación ante errores. En la transmisión fuera de banda, las unidades de NAL del conjunto de parámetros se transmiten en un canal diferente al de las otras unidades de NAL.

**[0193]** La norma de Sistemas MPEG-2 admite extensiones del sistema mediante "descriptores". Tanto las PMT como los PSM incluyen bucles de descriptores en los que se pueden insertar uno o más descriptores. En general, un descriptor puede comprender una estructura de datos que puede usarse para extender la definición de programas y/o elementos de programas. Esta divulgación describe los descriptores de un punto operativo para realizar las técnicas de esta divulgación. En general, el descriptor de punto operativo de esta divulgación mejora el descriptor convencional de la extensión de la MVC, describiendo una capacidad de representación, una capacidad de decodificación y una velocidad de bits para un punto operativo. Un dispositivo de destino, tal como el dispositivo de destino de A/V 140, puede usar descriptores de punto operativo para cada punto operativo, para seleccionar uno de los puntos operativos de un flujo de bits, para ser decodificado.

**[0194]** Cada PMT o PSM puede incluir un descriptor de punto operativo que describe las características de un punto operativo. Por ejemplo, el dispositivo de origen 120 puede proporcionar el descriptor de punto operativo para proporcionar un valor de capacidad de representación que describe una capacidad de representación para el

dispositivo cliente / de destino 140. Para que el dispositivo cliente 140 pueda representar adecuadamente (por ejemplo, mostrar) datos de vídeo del punto operativo, el dispositivo cliente 140 debería satisfacer las capacidades de representación señalizadas por el valor de la capacidad de representación. El valor de la capacidad de representación puede describir, por ejemplo, una serie de vistas que se mostrarán (por ejemplo, una serie de vistas destinadas a la representación) y/o la velocidad de tramas de los datos de vídeo para las vistas. Por lo tanto, el dispositivo cliente 140 puede determinar que las capacidades de representación se satisfacen cuando la salida de vídeo 144 del dispositivo cliente 140 puede mostrar el número de vistas del punto operativo a la velocidad de tramas especificada por el descriptor del punto operativo.

**[0195]** Después de que el multiplexor 30 haya ensamblado una unidad de NAL y/o una unidad de acceso a partir de los datos recibidos, el multiplexor 30 pasa la unidad a la interfaz de salida 132 para la salida. La interfaz de salida 132 puede comprender, por ejemplo, un transmisor, un transceptor, un dispositivo para escribir datos en un medio legible por ordenador, tal como, por ejemplo, una unidad óptica, una unidad de medios magnéticos (por ejemplo, una unidad de disquete), un puerto del bus universal en serie (USB), una interfaz de red u otra interfaz de salida. La interfaz de salida 132 emite la unidad de NAL o la unidad de acceso a un medio legible por ordenador 134, tal como, por ejemplo, un canal de transmisión, un medio magnético, un medio óptico, una memoria, una unidad flash u otro medio legible por ordenador.

**[0196]** En última instancia, la interfaz de entrada 136 recupera los datos desde un medio legible por ordenador 134. La interfaz de entrada 136 puede comprender, por ejemplo, una unidad óptica, una unidad de medios magnéticos, un puerto del USB, un receptor, un transceptor u otra interfaz de medio legible por ordenador. La interfaz de entrada 136 puede proporcionar la unidad de NAL o la unidad de acceso al demultiplexor 138. El demultiplexor 138 puede demultiplexar un flujo de transporte o un flujo de programa en flujos de PES constituyentes, desempaquetar los flujos de PES para recuperar datos codificados y enviar los datos codificados al decodificador de audio 146 o al decodificador de vídeo 148, en función de si los datos codificados son parte de un flujo de audio o de vídeo, por ejemplo, según lo indicado por los encabezados de paquetes de PES del flujo. El decodificador de audio 146 decodifica los datos de audio codificados y envía los datos de audio decodificados a la salida de audio 142, mientras que el decodificador de vídeo 148 decodifica los datos de vídeo codificados y envía los datos de vídeo decodificados, que pueden incluir una pluralidad de vistas de un flujo, a la salida de vídeo 144. La salida de vídeo 144 puede comprender una pantalla que utiliza una pluralidad de vistas de una escena, por ejemplo, una pantalla estereoscópica o autoestereoscópica que presenta cada vista de una escena simultáneamente.

**[0197]** En particular, el demultiplexor 138 puede seleccionar un punto operativo de un flujo de bits recibido. Por ejemplo, el demultiplexor 138 puede comparar las características de puntos operativos del flujo de bits para seleccionar un punto operativo adecuado para ser utilizado por el dispositivo de destino de A/V 140. En general, el demultiplexor 138 puede intentar seleccionar uno de los puntos operativos, que proporcionará la experiencia de visualización de la más alta calidad para un usuario, que pueda ser decodificado por el decodificador de vídeo 148. Por ejemplo, el demultiplexor 138 puede comparar las capacidades de representación y decodificación del decodificador de vídeo 148 con las capacidades sugeridas de representación y decodificación señalizadas por los descriptores de punto operativo del flujo de bits. De los puntos operativos que el demultiplexor 138 determina que podrían ser decodificados correctamente por el decodificador de vídeo 148, el demultiplexor 138 puede seleccionar un punto operativo que proporcionará datos de vídeo de la más alta calidad, por ejemplo, la máxima velocidad de tramas y/o velocidad de bits. En otros ejemplos, el demultiplexor 138 puede seleccionar uno de los puntos operativos con soporte basándose en otras consideraciones, tales como, por ejemplo, el consumo de energía.

**[0198]** En general, el sistema 100 puede corresponder esencialmente al sistema 10 de la figura 1. De manera similar, el multiplexor 130 puede corresponder esencialmente al multiplexor 21 de la figura 1, el demultiplexor 138 puede corresponder esencialmente al demultiplexor 29 de la figura 1, y otros componentes del sistema 100 con nombres similares pueden corresponder esencialmente a los componentes con nombres similares de la figura 1. Por lo tanto, el multiplexor 130 y el demultiplexor 138 pueden configurarse para realizar cualquiera de las diversas técnicas descritas en esta divulgación, solos o en cualquier combinación.

**[0199]** A continuación se describen algunos ejemplos de casos de uso. El uso del descriptor de jerarquía y del descriptor de extensión de jerarquía se pueden ilustrar en el siguiente ejemplo. La figura 5 es un diagrama conceptual que ilustra un ejemplo que ilustra flujos de bits de la L-HEVC transportados en cuatro flujos elementales. Por ejemplo, dado un programa que contiene flujos de bits de la L-HEVC transportados en cuatro flujos elementales, como se muestra en la figura 5, la información de dependencia para cada flujo elemental se señala de la siguiente manera:

- a. Un descriptor de jerarquía estará presente para *esA*, para describir que *esA* es un flujo elemental de capa base.
- b. Un descriptor de jerarquía estará presente para *esB*, para describir que *esB* tiene dependencia temporal de *esA*.
- c. Un descriptor de extensión de jerarquía estará presente para *esC*, para describir que *esC* tiene dependencia por capa de *esA*.

d. Un descriptor de extensión de jerarquía estará presente para *esD*, para describir que:

- i. *esD* tiene dependencia por capa de *esB*.
- ii. *esD* tiene dependencia temporal de *esC*.

**[0200]** En una alternativa, o agregado, para evitar el uso solapado del descriptor de jerarquía y del descriptor de extensión de jerarquía, se propone lo siguiente:

a. Para cada flujo elemental que lleva la imagen codificada con HEVC, un descriptor (por ejemplo, exactamente uno), ya sea el descriptor de jerarquía o el descriptor de extensión de jerarquía, puede estar (en un ejemplo, estará) presente y asociado al flujo elemental.

b. Un descriptor de jerarquía puede estar (en un ejemplo, estará) presente y asociado a un flujo elemental si el flujo elemental tiene dependencia (por ejemplo, temporal, espacial, de vista múltiple, etc.) de exactamente otro flujo elemental de referencia. Un flujo elemental de referencia de un flujo elemental actual es un flujo elemental que depende directamente del flujo elemental actual.

c. Un descriptor de extensión de jerarquía puede estar (en un ejemplo, estará) presente y asociado a un flujo elemental si el flujo elemental depende de más de un flujo elemental de referencia.

**[0201]** La figura 6 es otro ejemplo de dependencia entre flujos elementales. Como la figura 5, se ilustran los flujos de bits de L-HEVC transportados en cuatro flujos elementales. El uso del descriptor de jerarquía y del descriptor de extensión de jerarquía en este ejemplo alternativo o adicional para la estructura de dependencia, como se muestra en la figura 5, es el siguiente:

- a. Un descriptor de jerarquía estará presente para *esA*, para describir que *esA* es un flujo elemental de capa base.
- b. Un descriptor de jerarquía estará presente para *esB*, para describir que *esB* tiene dependencia temporal de *esA*.
- c. Un descriptor de jerarquía estará presente para *esC*, para describir que *esC* tiene dependencia de capa de *esA*.
- d. Un descriptor de extensión de jerarquía estará presente para *esD*, para describir que:

- iii. *esD* tiene dependencia de capa de *esB*
- iv. *esD* tiene dependencia temporal de *esC*.

**[0202]** Para la estructura de dependencia dada en la figura 6, el uso del descriptor de jerarquía y del descriptor de extensión de jerarquía en esta alternativa es el siguiente:

- a. Un descriptor de jerarquía estará presente para *esA*, para describir que *esA* es un flujo elemental de capa base.
- b. Un descriptor de jerarquía estará presente para *esB*, para describir que *esB* tiene dependencia temporal de *esA*.
- c. Un descriptor de jerarquía estará presente para *esC*, para describir que *esC* depende de la capa de *esB*.
- d. Un descriptor de jerarquía estará presente para *esD*, para describir que *esD* depende de la capa de *esC*.
- e. Un descriptor de jerarquía estará presente para *esE*, para describir que *esE* tiene dependencia temporal de *esD*.

**[0203]** La figura 7 ilustra otro ejemplo de la estructura de dependencia. Para la estructura de dependencia dada en la figura 7 (posiblemente la figura 6), el uso del descriptor de jerarquía y del descriptor de extensión de jerarquía en esta alternativa es el siguiente:

- a. Un descriptor de jerarquía estará presente para *esA*, para describir que *esA* es un flujo elemental de capa base.
- b. Un descriptor de jerarquía estará presente para *esB*, para describir que *esB* tiene dependencia temporal de *esA*.
- c. Un descriptor de jerarquía estará presente para *esC*, para describir que *esC* es un flujo elemental con codificación independiente.
- d. Un descriptor de jerarquía estará presente para *esD*, para describir que *esD* tiene dependencia temporal de *esC*.
- e. Un descriptor de extensión de jerarquía estará presente para *esE*, para describir que:

- i. *esE* tiene dependencia de capa de *esC*
- ii. *esE* tiene dependencia de capa de *esA*.

f. Un descriptor de extensión de jerarquía estará presente para *esF*, para describir que:

- iii. *esF* tiene dependencia temporal de *esE*
- iv. *esF* tiene dependencia de capa de *esD*.
- v. *esF* tiene dependencia de capa de *esB*.

**[0204]** En la segunda alternativa o ejemplo adicional para evitar el uso solapado del descriptor de jerarquía y del descriptor de extensión de jerarquía, se propone lo siguiente:

- a. Un descriptor de jerarquía estará presente y asociado a un flujo elemental si el flujo elemental tiene dependencia temporal.
- b. Un descriptor de jerarquía estará presente y asociado a un flujo elemental si el flujo elemental tiene dependencia de capa (por ejemplo, espacial, de múltiples vistas, de calidad) de exactamente otro flujo elemental de referencia.
- c. Un descriptor de extensión de jerarquía estará presente y asociado a un flujo elemental si el flujo elemental tiene dependencia de capa de más de un flujo elemental de referencia.

**[0205]** El uso del descriptor de jerarquía y del descriptor de extensión de jerarquía en este ejemplo alternativo o adicional para la estructura de dependencia, como se muestra en la figura 5 (y posiblemente las figuras 6 o 7), es el siguiente:

- a. Un descriptor de jerarquía estará presente para *esA*, para describir que *esA* es un flujo elemental de capa base.
- b. Un descriptor de jerarquía estará presente para *esB*, para describir que *esB* tiene dependencia temporal de *esA*.
- c. Un descriptor de jerarquía estará presente para *esC*, para describir que *esC* tiene dependencia de capa de *esA*.
- d. Estarán presentes dos descriptores de jerarquía para *esD*, para describir que *esD* tiene dependencia temporal de *esC* y que *esD* tiene dependencia de capa de *esB*.

**[0206]** El uso del descriptor de jerarquía y del descriptor de extensión de jerarquía en esta alternativa para la estructura de dependencia, como se muestra en la figura 7 (posiblemente las figuras 5 o 6), es el siguiente:

- a. Un descriptor de jerarquía estará presente para *esA*, para describir que *esA* es un flujo elemental de capa base.
- b. Un descriptor de jerarquía estará presente para *esB*, para describir que *esB* tiene dependencia temporal de *esA*.
- c. Un descriptor de jerarquía estará presente para *esC*, para describir que *esC* es un flujo elemental con codificación independiente.
- d. Un descriptor de jerarquía estará presente para *esD*, para describir que *esD* tiene dependencia temporal de *esC*.
- e. Un descriptor de extensión de jerarquía estará presente para *esE*, para describir que:
  - v. *esE* tiene dependencia de capa de *esC*
  - vi. *esE* tiene dependencia de capa de *esA*.

- f. Un descriptor de jerarquía estará presente para *esF*, para describir que *esF* tiene dependencia temporal de *esE*.
- g. Un descriptor de extensión de jerarquía estará presente para *esF*, para describir que:

- vii. *esF* tiene dependencia de capa de *esD*
- viii. *esF* tiene dependencia de capa de *esB*

**[0207]** En el tercer ejemplo alternativo o adicional para evitar el uso solapado del descriptor de jerarquía y del descriptor de extensión de jerarquía, se propone que solo se use el descriptor de jerarquía, y que se pueda eliminar el descriptor de extensión de jerarquía.

**[0208]** El uso del descriptor de jerarquía en esta alternativa para la estructura de dependencia, como se muestra en la figura 5 (posiblemente las figuras 6 o 7), es el siguiente:

- a. Un descriptor de jerarquía estará presente para *esA*, para describir que *esA* es un flujo elemental de capa base.

- b. Un descriptor de jerarquía estará presente para *esB*, para describir que *esB* tiene dependencia temporal de *esA*.
- c. Un descriptor de jerarquía estará presente para *esC*, para describir que *esC* tiene dependencia de capa de *esA*.
- d. Un descriptor de jerarquía estará presente para *esD*, para describir que *esD* tiene dependencia temporal de *esC*.
- e. Un descriptor de jerarquía estará presente para *esD*, para describir que *esD* tiene dependencia de capa de *esB*.

**[0209]** Ante la presencia del descriptor de jerarquía o del descriptor de extensión de jerarquía, o de ambos, se proponen las dos siguientes opciones:

- a. Opción 1: para cada flujo elemental que contiene imágenes codificadas con HEVC, al menos un descriptor que contenga información de dependencia estará presente.
- b. Opción 2:
  - i. La presencia de un descriptor que contenga información de dependencia es obligatoria solo cuando un programa contiene más de un flujo elemental de imágenes codificadas con HEVC con el mismo tipo de flujo.
  - ii. Cuando ningún descriptor que contenga información de dependencia está presente en un programa, para cada flujo elemental en el programa que contiene imágenes codificadas con HEVC, el valor de *índice\_capa\_jerarquía* para cada uno de los flujos elementales se obtiene de la siguiente manera: se da al primer flujo elemental el valor 0, se da al segundo flujo elemental el valor 1, y se da al n-ésimo flujo elemental el valor n - 1.

**[0210]** El uso del descriptor de jerarquía y del descriptor de extensión de jerarquía en el ejemplo del segundo ejemplo alternativo o adicional para la estructura de dependencia, como se muestra en la figura 5 (posiblemente las Figuras 6 o 7), no necesita ni descriptor de jerarquía ni descriptor de extensión de jerarquía. Como cada uno de los flujos elementales es de diferentes tipos, no es necesario señalar el descriptor para la información de dependencia, ya que se puede deducir de la siguiente manera:

- a. *esA* es un flujo elemental de capa base
- b. *esB* tiene dependencia temporal de *esA*
- c. *esC* tiene dependencia de múltiples vistas de *esA*
- d. *esD* tiene tanto dependencia de múltiples vistas de *esC* como dependencia temporal de *esB*.

**[0211]** Además, el orden de esos flujos elementales en la tabla de mapas del programa es el siguiente: *esA*, *esB*, *esC* y *esD*.

**[0212]** Por último, el valor de *índice\_capa\_jerarquía* para cada flujo elemental se obtiene de la siguiente manera: el *índice\_capa\_jerarquía* de *esA* es igual a 0, el *índice\_capa\_jerarquía* de *esB* es igual a 1, el *índice\_capa\_jerarquía* de *esC* es igual a 2 y el *índice\_capa\_jerarquía* de *esD* es igual a 4

**[0213]** Los siguientes son ejemplos de implementaciones de una o más técnicas ejemplares descritas anteriormente. Se proporcionan los textos sugeridos para implementar las propuestas anteriores.

**[0214]** Para un descriptor de extensión de jerarquía mejorado, la semántica de *bits\_dimensión\_extensión* se actualiza de la siguiente manera. El texto en cursiva representa las actualizaciones para la norma MPEG-2 TS:

**[0215]** ***bits\_dimensión\_extensión***: un campo de 16 bits que indica la posible mejora del elemento de programa asociado proveniente de la capa base, resultante del elemento de programa de la capa con el identificador de capa *nuh* igual a 0.

**[0216]** La asignación de los bits a las dimensiones de mejora es la siguiente en la Tabla Amd7-4.

**Tabla Amd7-4 - Semántica de bits de dimensión de extensión**

Índice para bits	Descripción
0	Mejora de vista múltiple
1	Ajustabilidad espacial a escala, incluyendo la SNR
2	Mejora de profundidad
3	Capa base de la AVC
4	Capa base de MPEG-2
5	<i>Mejora temporal</i>

Índice para bits	Descripción
6~15	Reservado

[0217] Téngase en cuenta que el valor del índice para los bits asignados para la mejora temporal puede ser cualquiera de los valores reservados (por ejemplo, 5 a 15).

5 [0218] Para un descriptor de jerarquía mejorado, se proponen los siguientes cambios para la implementación del ejemplo para el soporte de la mejora de la capa auxiliar en el descriptor de jerarquía. Téngase en cuenta que la parte actualizada está resaltada en gris, indicándose con [[]] la eliminación:

10 **Cláusula 2.6.6**

[0219] Reemplace la Tabla 2-49 por:

**Tabla 2-49 - Descriptor de jerarquía**

Sintaxis	Nº de bits	Mnemónico
descriptor_jerarquía () {		
<b>etiqueta_descriptor</b>	<b>8</b>	<b>uimsbf</b>
<b>longitud del descriptor</b>	<b>8</b>	<b>uimsbf</b>
<b>indicador_sin_ajustabilidad_escalavista</b>	<b>1</b>	<b>bslbf</b>
<b>indicador_sin_ajustabilidad_temporal_escalavista</b>	<b>1</b>	<b>bslbf</b>
<b>indicador_sin_ajustabilidad_espacial_escalavista</b>	<b>1</b>	<b>bslbf</b>
<b>indicador_sin_ajustabilidad_calidad_escalavista</b>	<b>1</b>	<b>bslbf</b>
<b>tipo_jerarquía</b>	<b>4</b>	<b>uimsbf</b>
<i>indicador_sin_auxiliar</i>	<i>1</i>	<i>bslbf</i>
<b>reservado</b>	<b>1</b> [eliminado: "2"]	<b>bslbf</b>
<b>índice_capa_jerarquía</b>	<b>6</b>	<b>uimsbf</b>
<b>indicador_tref_presente</b>	<b>1</b>	<b>bslbf</b>
<b>reservado</b>	<b>1</b>	<b>bslbf</b>
<b>índice_capa_integrada_jerarquía</b>	<b>6</b>	<b>uimsbf</b>
<b>reservado</b>	<b>2</b>	<b>bslbf</b>
<b>canal_jerarquía</b>	<b>6</b>	<b>uimsbf</b>
}		

15 **Cláusula 2.6.7**

**Reemplazar:**

20 [0220] **indicador\_ajustabilidad\_temporal\_escalavista**: un indicador de 1 bit que, cuando se fija en '0', indica que el elemento de programa asociado mejora la velocidad de tramas del flujo de bits resultante del elemento de programa al que hace referencia el índice\_capa\_integrada\_jerarquía. El valor de '1' para este indicador está reservado.

25 [0221] **indicador\_ajustabilidad\_espacial\_escalavista**: un indicador de 1 bit que, cuando se fija en '0', indica que el elemento de programa asociado mejora la resolución espacial del flujo de bits resultante del elemento de programa al que hace referencia el índice\_capa\_integrada\_jerarquía. El valor de '1' para este indicador está reservado.

30 [0222] **indicador\_ajustabilidad\_calidad\_escalavista**: un indicador de 1 bit que, cuando se fija en '0', indica que el elemento de programa asociado mejora la calidad o fidelidad de la SNR del flujo de bits resultante del elemento de programa al que hace referencia el índice\_capa\_integrada\_jerarquía. El valor de '1' para este indicador está reservado.

[0223] **tipo\_jerarquía**: la relación jerárquica entre la capa jerárquica asociada y su capa integrada de jerarquía se define en la Tabla 2-50. Si la ajustabilidad a escala se aplica en más de una dimensión, este campo se fijará en el valor de '8' ("Ajustabilidad a escala combinada"), y los indicadores indicador\_ajustabilidad\_temporal\_escalavista, indicador\_ajustabilidad\_espacial\_escalavista e indicador\_ajustabilidad\_calidad\_escalavista se fijarán en consecuencia. Para

los subflujos de bits de vídeo de la MVC, este campo se fijará en el valor de '9' ("Subflujo de bits de vídeo de MVC") y los indicadores indicador\_ajustabilidad\_temporal\_escalas, indicador\_ajustabilidad\_espacial\_escalas e indicador\_ajustabilidad\_calidad\_escalas se fijarán en '1'. Para los subflujos de bits de la vista base de la MVC, este campo se fijará en el valor de '15' y los indicadores indicador\_ajustabilidad\_temporal\_escalas, indicador\_ajustabilidad\_espacial\_escalas e indicador\_ajustabilidad\_calidad\_escalas se fijarán en '1'. Para los subflujos de bits de vídeo de MVCD, este campo se fijará en el valor de '9' ("Subflujo de bits de vídeo de MVCD") y los indicadores indicador\_ajustabilidad\_temporal\_escalas, indicador\_ajustabilidad\_espacial\_escalas e indicador\_ajustabilidad\_calidad\_escalas se fijarán en '1'. Para los subflujos de bits de la vista base de MVCD, este campo se fijará en el valor de '15' y los indicadores indicador\_ajustabilidad\_temporal\_escalas, indicador\_ajustabilidad\_espacial\_escalas e indicador\_ajustabilidad\_calidad\_escalas se fijarán en '1'.

con:

**[0224] indicador\_sin\_ajustabilidad\_vista\_escalas:** un indicador de 1 bit que, cuando se fija en '0', indica que el elemento de programa asociado mejora el número de vistas del flujo de bits resultante del elemento de programa al que hace referencia el índice\_capa\_integrada\_jerarquía. El valor de '1' para este indicador está reservado.

**[0225] indicador\_sin\_ajustabilidad\_temporal\_escalas:** un indicador de 1 bit que, cuando se fija en '0', indica que el elemento de programa asociado mejora la velocidad de tramas del flujo de bits resultante del elemento de programa al que hace referencia el índice\_capa\_integrada\_jerarquía. El valor de '1' para este indicador está reservado.

**[0226] indicador\_sin\_ajustabilidad\_espacial\_escalas:** un indicador de 1 bit que, cuando se fija en '0', indica que el elemento de programa asociado mejora la resolución espacial del flujo de bits resultante del elemento de programa al que hace referencia el índice\_capa\_integrada\_jerarquía. El valor de '1' para este indicador está reservado.

**[0227] indicador\_sin\_ajustabilidad\_calidad\_escalas:** un indicador de 1 bit que, cuando se fija en '0', indica que el elemento de programa asociado mejora la calidad de la SNR o la fidelidad del flujo de bits resultante del elemento de programa al que hace referencia el índice\_capa\_integrada\_jerarquía. El valor de '1' para este indicador está reservado.

**[0228] tipo\_jerarquía:** la relación jerárquica entre la capa jerárquica asociada y su capa integrada de jerarquía se define en la Tabla 2-50. Si la ajustabilidad a escala se aplica en más de una dimensión, este campo se fijará en el valor de '8' ("Ajustabilidad a escala combinada") y los indicadores indicador\_sin\_ajustabilidad\_vista\_escalas, indicador\_sin\_ajustabilidad\_temporal\_escalas, indicador\_sin\_ajustabilidad\_espacial\_escalas e indicador\_sin\_ajustabilidad\_calidad\_escalas se configurarán en consecuencia. Para los subflujos de bits de vídeo de la MVC, este campo se fijará en el valor de '9' ("Subflujo de bits de vídeo de MVC") y los indicadores indicador\_sin\_ajustabilidad\_vista\_escalas, indicador\_sin\_ajustabilidad\_temporal\_escalas, indicador\_sin\_ajustabilidad\_espacial\_escalas e indicador\_sin\_ajustabilidad\_calidad\_escalas se fijarán en '1'. Para subflujos de bits de vista base de MVC, este campo se fijará en el valor de '15' y los indicadores indicador\_sin\_ajustabilidad\_vista\_escalas, indicador\_sin\_ajustabilidad\_temporal\_escalas, indicador\_sin\_ajustabilidad\_espacial\_escalas, [[and]] indicador\_sin\_ajustabilidad\_calidad\_escalas se fijarán en '1' y *el indicador\_sin\_auxiliar se fijará en '1'*. Para los subflujos de bits de vídeo de MVCD, este campo se fijará en el valor de '9' ("Subflujo de bits de vídeo de MVCD") y los indicadores indicador\_sin\_ajustabilidad\_vista\_escalas, indicador\_sin\_ajustabilidad\_temporal\_escalas, indicador\_sin\_ajustabilidad\_espacial\_escalas e indicador\_sin\_ajustabilidad\_calidad\_escalas se fijarán en '1'. Para los subflujos de bits de vista base de MVCD, este campo se fijará en el valor de '15' y los indicadores indicador\_sin\_ajustabilidad\_vista\_escalas, indicador\_sin\_ajustabilidad\_temporal\_escalas, indicador\_sin\_ajustabilidad\_espacial\_escalas, [[and]] indicador\_sin\_ajustabilidad\_calidad\_escalas se fijarán en '1' y *el indicador\_sin\_auxiliar se fijará en '1'*.

**[0229] indicador\_sin\_auxiliar:** un indicador de 1 bit que, cuando se fija en '0', indica que el elemento de programa asociado proporciona una mejora auxiliar resultante del elemento de programa al que hace referencia el índice\_capa\_integrada\_jerarquía. El valor de '1' para este indicador está reservado.

**[0230]** Reemplazar en la Tabla 2-50 la descripción para el valor 15 de la siguiente manera:

**Tabla 2-50 - valores del campo tipo\_jerarquía**

Valor	Descripción
10	Auxiliar
15	Subflujo de bits de capa base o de vista base de la MVC o subflujo de bits de vídeo de AVC de la MVC o subflujo de bits de vídeo temporal de la HEVC o subpartición de base de la HEVC.

**[0231]** Para el uso propuesto del descriptor de jerarquía y del descriptor de extensión de jerarquía, se proponen los siguientes cambios para la implementación de la evitación del uso solapado del descriptor de jerarquía y del descriptor de extensión de jerarquía.



**[0232]** Agregar al final de la lista con viñetas en la sección 2.17.1 del borrador de L-HEVC TS:

- a. Un descriptor de jerarquía estará presente para cada flujo elemental con tipo\_flujo igual a 0x24 y 0x25.
- b. Un descriptor de extensión de jerarquía estará presente para cada flujo elemental con tipo\_flujo igual a 0x27, 0x28, 0x29 y 0x2A.

**[0233]** Se proponen los siguientes cambios para la implementación de la evitación del solapado.

**[0234]** Agregar al final de la lista con viñetas en la sección 2.17.1 del borrador de L-HEVC TS:

- a. Un descriptor de jerarquía estará presente para cada flujo elemental con tipo\_flujo igual a 0x24 y 0x25.
- b. Para cada flujo elemental con tipo\_flujo igual a 0x27, 0x28, 0x29 y 0x2A, se aplica lo siguiente:
  - i. Si el flujo elemental mejora exactamente otro flujo elemental, estará presente un descriptor de jerarquía.
  - ii. De lo contrario, estará presente un descriptor de extensión de jerarquía

**[0235]** Se describe ahora la segunda alternativa o ejemplo adicional del uso propuesto del descriptor de jerarquía y del descriptor de extensión de jerarquía. Se proponen los siguientes cambios para la implementación de la evitación del solapado.

**[0236]** Agregar al final de la lista con viñetas en la sección 2.17.1 del borrador de L-HEVC TS:

- a. Para el flujo elemental con tipo\_flujo igual a 0x24, estará presente un descriptor de jerarquía con tipo\_jerarquía igual a 15.
- b. Para cada flujo elemental con tipo\_flujo igual a 0x25, 0x28 y 0x29, estará presente un descriptor de jerarquía con tipo\_jerarquía igual a 3.
- c. Para cada flujo elemental con tipo\_flujo igual a 0x27, 0x28, 0x29 y 0x2A, se aplica lo siguiente:
  - i. Si el flujo elemental mejora exactamente otro flujo elemental, estará presente un descriptor de jerarquía con tipo\_jerarquía no igual a 3.
  - ii. De lo contrario, estará presente un descriptor de extensión de jerarquía

**[0237]** La tercera alternativa o adición del uso propuesto del descriptor de jerarquía y del descriptor de extensión de jerarquía. Se proponen los siguientes cambios para la implementación de la evitación del solapado.

**[0238]** Agregar al final de la lista con viñetas en la sección 2.17.1 del borrador de L-HEVC TS:

- a. Para cada flujo elemental con tipo\_flujo igual a 0x24 y 0x25, estará presente un descriptor de jerarquía.
- b. Para cada flujo elemental con tipo\_flujo igual a 0x27, 0x28, 0x29 o 0x2A, estarán presentes uno o más descriptores de jerarquía.

**[0239]** Adicional o alternativamente, un descriptor de vídeo de la HEVC puede ser obligatorio para cada flujo elemental con tipo\_flujo igual a 0x24 y 0x25. Del mismo modo, cuando un programa del documento Rec. UIT-T H.222,0 | ISO/IEC 13818-1 incluye uno o más flujos elementales con *tipo\_flujo* igual a 0x27, 0x28, 0x29 o 0x2A, por lo menos un descriptor de punto operativo de la HEVC puede ser obligatorio en la tabla de mapas de programa asociada al programa.

**[0240]** La figura 8 es un diagrama conceptual que ilustra un ejemplo de múltiples capas de datos de vídeo. Un ejemplo en el que la información del punto operativo puede ser señalizada con un descriptor de punto operativo de la HEVC, por ejemplo, el descriptor de punto operativo de la HEVC de la Tabla propuesta Amd7-1, como se ha expuesto anteriormente, se describe con respecto a los datos de vídeo de la figura 8. La figura 8 representa un ejemplo de un programa que tiene cinco capas de un flujo de bits en capas de la HEVC, dividido en cinco flujos elementales. La dependencia de cada flujo elemental es como se muestra en la figura 8 mediante flechas entre las capas. La dependencia puede ser descrita por los datos de un descriptor de extensión de jerarquía, en algunos ejemplos.

**[0241]** Debido al número limitado de octetos que puede tener un descriptor, la señalización de los puntos operativos se puede dividir en dos descriptores de punto operativo de la HEVC, como se muestra a continuación. El primer descriptor de punto operativo de la HEVC, en este ejemplo, contiene información de las tres primeras estructuras de PTL y de los dos primeros puntos operativos, mientras que el segundo descriptor de punto operativo de la HEVC contiene información de las dos siguientes estructuras de PTL y de los dos siguientes puntos operativos.

**Descriptor del primer punto operativo**

**[0242]**

```

5      #PTL=3
      PTL [0] = Perfil principal, grada principal, nivel 3.1
      PTL [1] = Perfil principal de MV, grada principal, nivel 3.1
      PTL [2] = Perfil principal de MV, grada principal, nivel 4.1
      #OP = 2
      recuento_es [0] = 1
10     dependencias_antepuestas[0] = 1
      referencia_es [0] [0] = 1
      #capas [0] = 2
      idx_ref_pt1 [0] [0] = 0
      idx_ref_pt1 [0] [1] = 1
15     recuento_es [1] = 1
      dependencias_antepuestas[0] = 1
      referencia_es [1] [0] = 2
      #capas [1] = 3
      idx_ref_pt1 [1] [0] = 0
20     idx_ref_pt1 [1] [1] = 1
      idx_ref_pt1 [1] [2] = 2

```

**Descriptor del segundo punto operativo****[0243]**

```

25     #PTL=1
      PTL [3] = Perfil principal de MV, grada principal, nivel 5.0
      #OP = 2
30     recuento_es [2] = 1
      dependencias_antepuestas [2] = 0
      referencia_es [2] [0] = 3
      #capas [2] = 1
      idx_ref_pt1 [2] [0] = 3
35     recuento_es [3] = 2
      dependencias_antepuestas [3] = 1
      referencia_es [3] [0] = 4
      #capas [3] = 5
      idx_ref_pt1 [3] [0] = 0
40     idx_ref_pt1 [3] [1] = 1
      idx_ref_pt1 [3] [2] = 2
      idx_ref_pt1 [3] [3] = 0
      idx_ref_pt1 [3] [4] = 1

```

45 **[0244]** Téngase en cuenta que, para ahorrar bits, para algunos puntos operativos, el valor de *dependencias\_antepuestas[op/idx]* se fija igual a 1 y la lista de flujos elementales del punto operativo se puede obtener en función de la información de dependencia señalizada en el descriptor de extensión de jerarquía.

50 **[0245]** La figura 9 es un diagrama de flujo que ilustra un procedimiento ejemplar de acuerdo a las técnicas de esta divulgación. En este ejemplo, el procedimiento de la figura 9 se explica con respecto al demultiplexor 29 de la figura 1. Sin embargo, debería entenderse que otros dispositivos pueden configurarse para realizar el procedimiento de la figura 9, tal como el demultiplexor 138 de la figura 4, o un elemento de red consciente de los medios (MANE), situado entre un dispositivo de origen y un dispositivo de destino. Además, un multiplexor puede realizar un procedimiento recíproco, tal como el multiplexor 21 de la figura 1 o el multiplexor 130 de la figura 4.

55 **[0246]** Inicialmente, en este ejemplo, el demultiplexor 29 extrae un descriptor, tal como un descriptor de punto operativo de la HEVC de acuerdo a la Tabla Amd7-1 propuesta, de un flujo de bits (150). El demultiplexor 29 recupera luego una o más estructuras de perfil, grada y nivel (PTL) del descriptor (152). Por ejemplo, las estructuras de PTL pueden corresponder a los datos del bucle "para (i = 0; i < núm\_perfil\_grada\_nivel; i ++, ptIdx ++)" de la Tabla Amd7-1 propuesta que se ha expuesto anteriormente.

60 **[0247]** El demultiplexor 29 puede determinar dependencias entre flujos elementales (es decir, las capas) (154). Las dependencias se pueden señalar en el descriptor del punto operativo de la HEVC o en un descriptor distinto. Por ejemplo, el demultiplexor 29 puede extraer un descriptor de jerarquía (por ejemplo, de acuerdo a la Tabla 2-49) o un descriptor de extensión de jerarquía que incluye información indicativa de las dependencias, como se ha expuesto anteriormente.

- 5 **[0248]** El demultiplexor 29 también determina cuál de las estructuras de PTL corresponde a cada capa de cada uno entre una pluralidad de puntos operativos (156). Por ejemplo, el demultiplexor 29 puede determinar esta correspondencia a partir del bucle "para (j = 0; j < núm\_capas [opldx]; j ++)" de la Tabla Amd7-1 propuesta, expuesta anteriormente.
- 10 **[0249]** En algunos ejemplos, el demultiplexor 29 puede iterar por las estructuras de PTL para determinar la información de PTL de cada una de las estructuras de PTL, por ejemplo, los valores para idc\_perfil [ptlldx], indicador\_grada [ptlldx] e idc\_nivel [ptlldx]. El demultiplexor 29 puede formar estructuras de datos representativas de cada una de las estructuras de PTL señalizadas en el descriptor (es decir, cada una de las estructuras de PTL correspondientes a uno de los valores de ptlldx). El demultiplexor 29 puede además recuperar valores para los índices de referencia de PTL para cada una de las capas de cada uno de los puntos operativos que correlacionan una capa correspondiente de las capas con una de las estructuras de PTL, para determinar el perfil, la grada y la información de nivel para la capa correspondiente de las capas. Por ejemplo, idx\_ref\_ptl[opldx][j] de la Tabla Amd7-1 propuesta
- 15 representa un ejemplo de un elemento sintáctico que tiene un valor que correlaciona la capa "j" del punto operativo "opldx" con una de las estructuras de PTL.
- 20 **[0250]** Además, el demultiplexor 29 puede determinar conjuntos de capas de salida para cada uno de los puntos operativos (158). Por ejemplo, el demultiplexor 29 puede determinar cuál de las capas se incluye en un conjunto de capas de salida para cada uno de los puntos operativos de los elementos sintácticos ols\_destino[opldx] de la Tabla Amd7-1 propuesta, expuesta anteriormente. En un ejemplo, el demultiplexor 29 recupera los valores para el elemento sintáctico ols\_destino[opldx], para cada uno de los puntos operativos del descriptor, que especifican los conjuntos de capas de salida de destino que están asociados a los puntos operativos correspondientes.
- 25 **[0251]** En base a la información del descriptor, el demultiplexor 29 selecciona un punto operativo (160). Por ejemplo, el demultiplexor 29 puede determinar uno de los puntos operativos que puede ser decodificado mediante el decodificador de vídeo 30. Un decodificador de vídeo puede decodificar un punto operativo cuando el decodificador de vídeo presta soporte a los elementos de perfil, grada y nivel a los que corresponden los datos de vídeo del punto operativo. Por lo tanto, si el decodificador de vídeo 30 presta soporte al menos al perfil, a la grada y al nivel del punto operativo, como lo indica la información de PTL del descriptor, el demultiplexor 29 puede seleccionar el punto operativo. El demultiplexor 29 puede basar además la selección en otras características, tales como un número de capas de salida de destino en un conjunto de capas de salida, por ejemplo, como lo indica la información del conjunto de capas de salida del descriptor para el punto operativo. En particular, el demultiplexor 29 puede determinar si el dispositivo de visualización 32 puede representar una serie de vistas igual al número de capas de salida de destino para el punto operativo. Si hay múltiples puntos operativos que pueden decodificarse y exhibirse, el demultiplexor 29 puede seleccionar un punto operativo con el perfil, grada y nivel máximos, que también incluya un número máximo de vistas que se pueden mostrar. Adicional o alternativamente, el demultiplexor 29 puede recibir entrada de un usuario, indicativa de un punto operativo a seleccionar.
- 30 **[0252]** De esta manera, en un ejemplo, el demultiplexor 29 puede determinar uno o más perfiles de una norma de codificación de vídeo a la que el decodificador de vídeo presta soporte, una o más gradas de la norma de codificación de vídeo a la que presta soporte el decodificador de vídeo, y uno o más niveles dentro de las una o más gradas a las que presta soporte el decodificador de vídeo, y luego seleccionar uno de los puntos operativos de tal manera que cada una de las capas de uno de los puntos operativos tenga uno de los perfiles a los que presta soporte el decodificador de vídeo, cada una de las capas de dicho uno de los puntos operativos tenga una de las gradas a las que presta soporte el decodificador de vídeo, y cada una de las capas de dicho uno de los puntos operativos tenga uno de los niveles a los que presta soporte el decodificador de vídeo, como lo indican las estructuras de PTL a las cuales corresponden las capas de dicho uno de los puntos operativos.
- 35 **[0253]** El demultiplexor 29 puede extraer luego los flujos elementales correspondientes al punto operativo seleccionado (162). Los flujos elementales a extraer pueden incluir flujos elementales que incluyen datos para cada uno de los conjuntos de capas de salida de destino, así como flujos elementales que incluyen datos de los que depende el conjunto de capas de salida de destino (por ejemplo, para la predicción temporal y/o entre vistas). El demultiplexor 29 puede enviar luego los flujos elementales extraídos al decodificador de vídeo 30 (164).
- 40 **[0254]** De esta manera, el procedimiento de la figura 9 representa un ejemplo de un procedimiento que incluye la extracción de un descriptor del flujo de bits, en el que el flujo de bits incluye capas de datos de vídeo para puntos operativos, independientes del descriptor, de manera que cada punto operativo incluya una o más de las capas de datos de vídeo, y en el que el descriptor incluye un conjunto de estructuras de perfil, grada y nivel (PTL) y datos que asocian cada una de las capas de cada uno de los puntos operativos a una estructura correspondiente de las estructuras de PTL, extrayendo los datos de vídeo para uno de los puntos operativos del flujo de bits, basándose, al menos en parte, en las estructuras de PTL a las que corresponden las capas de uno de los puntos operativos, y proporcionando los datos de vídeo extraídos a un decodificador de vídeo.
- 45 **[0255]** Se ha de reconocer que, según el ejemplo, ciertos actos o sucesos de cualquiera de las técnicas descritas en el presente documento pueden realizarse en una secuencia diferente, pueden agregarse, fusionarse u omitirse por
- 50
- 55
- 60
- 65

completo (por ejemplo, no todos los actos o sucesos descritos son necesario para la puesta en práctica de las técnicas). Además, en ciertos ejemplos, los actos o sucesos se pueden realizar de forma concurrente, por ejemplo, mediante el procesamiento de hilos múltiples, el procesamiento de interrupciones o procesadores múltiples, en lugar de secuencialmente.

5  
**[0256]** En uno o más aspectos, las funciones descritas pueden implementarse en hardware, software, firmware o cualquier combinación de los mismos. Si se implementan en software, las funciones, como una o más instrucciones o código, pueden almacenarse en, o transmitirse por, un medio legible por ordenador, y ser ejecutadas por una unidad de procesamiento basada en el hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que corresponden a un medio tangible tal como los medios de almacenamiento de datos, o medios de comunicación que incluyen cualquier medio que facilite la transferencia de un programa de ordenador desde un lugar a otro, por ejemplo, de acuerdo a un protocolo de comunicación. De esta manera, los medios legibles por ordenador generalmente pueden corresponder a (1) medios de almacenamiento tangibles legibles por ordenador que no son transitorios o (2) un medio de comunicación tal como una señal u onda portadora. Los medios de almacenamiento de datos pueden ser medios disponibles cualesquiera a los que puedan acceder 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.

20  
**[0257]** A modo de ejemplo, y no limitativo, tales medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otros dispositivos de almacenamiento en disco óptico, almacenamiento en disco magnético u otros tipos de almacenamiento magnético, memoria flash o cualquier otro medio que pueda ser utilizado para almacenar código de programa deseado en forma de instrucciones o estructuras de datos y al que se pueda acceder mediante un ordenador. Además, cualquier conexión se denomina correctamente un medio legible por ordenador. Por ejemplo, si el software es transmitido desde una sede en la Red, un servidor o cualquier 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 los infrarrojos, la radio o las microondas están incluidos 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 están orientados a medios de almacenamiento tangibles, no transitorios. Los discos, como se usan en el presente documento, incluyen discos compactos (CD), discos láser, discos ópticos, discos versátiles digitales (DVD), disquetes y discos Blu-ray®, donde algunos discos generalmente reproducen datos magnéticamente, mientras que otros discos reproducen datos ópticamente con láser. Las combinaciones de lo que antecede también deberían incluirse dentro del alcance de los medios legibles por ordenador.

35  
**[0258]** 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 lógica programable en el terreno (FPGA) u otros circuitos lógicos integrados o discretos equivalentes. Por consiguiente, el término "procesador", como se usa en el presente documento, puede referirse a 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 puede proporcionarse dentro de módulos de hardware y/o software dedicados, configurados para la codificación y decodificación, o incorporarse en un códec combinado. Además, las técnicas podrían implementarse completamente en uno o más circuitos o elementos lógicos.

45  
**[0259]** Las técnicas de esta divulgación pueden implementarse en una amplia variedad de dispositivos o aparatos, incluidos un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). Varios componentes, módulos o unidades se describen en esta divulgación para enfatizar los aspectos funcionales de los dispositivos configurados para realizar las técnicas divulgadas, pero no necesariamente requieren realización por diferentes unidades de hardware. Más bien, como se ha descrito anteriormente, varias unidades pueden combinarse en una unidad de hardware de códec o ser proporcionadas por una colección de unidades de hardware interoperativas, que incluyen uno o más procesadores como se ha descrito anteriormente, junto con el software y/o firmware adecuados.

50  
**[0260]** Se han descrito varios ejemplos. Estos y otros ejemplos están dentro del alcance de las siguientes reivindicaciones.

**REIVINDICACIONES**

1. Un procedimiento para procesar un flujo de bits que incluye datos de vídeo de la Codificación de vídeo de alta eficacia, HEVC, comprendiendo el procedimiento:

5 extraer un descriptor del flujo de bits (150), en donde el flujo de bits incluye capas de datos de vídeo para puntos operativos, independientes del descriptor, de manera que cada punto operativo incluya una o más de las capas de datos de vídeo, en donde la extracción del descriptor incluye:

10 extraer un conjunto de estructuras de perfil, grada y nivel (PTL); y

extraer datos que asocian cada una de las capas de cada uno de los puntos operativos con una estructura correspondiente de las estructuras de PTL, en donde los datos que asocian cada una de las capas con la estructura correspondiente de las estructuras de PTL son independientes de las estructuras de PTL en el descriptor;

15 extraer datos de vídeo para uno de los puntos operativos del flujo de bits (162) basándose, al menos en parte, en las estructuras de PTL a las que corresponden las capas de dicho uno de los puntos operativos;

20 y proporcionar los datos de vídeo extraídos a un decodificador de vídeo (164).

2. El procedimiento de la reivindicación 1, en el que la extracción de los datos de vídeo (162) comprende:

25 determinar uno o más perfiles de una norma de codificación de vídeo a la que presta soporte el decodificador de vídeo, una o más gradas de la norma de codificación de vídeo a la que presta soporte el decodificador de vídeo y uno o más niveles dentro de las una o más gradas a las que presta soporte el decodificador de vídeo; y

30 seleccionar dicho uno de los puntos operativos (160) de tal manera que cada una de las capas de dicho uno de los puntos operativos seleccionado tenga uno de los perfiles a los que presta soporte el decodificador de vídeo, cada una de las capas de dicho uno de los puntos operativos seleccionado tenga una de las gradas a las que presta soporte el decodificador de vídeo, y cada una de las capas de dicho uno de los puntos operativos seleccionado tenga uno de los niveles a los que presta soporte el decodificador de vídeo, según lo indicado por las estructuras de PTL a las que corresponden las capas de dicho uno de los puntos operativos.

3. El procedimiento de la reivindicación 1, que comprende además:

40 iterar por las estructuras de PTL para determinar al menos la información de perfil, la información de grada y la información de nivel para cada una de las estructuras de PTL; y, para cada una de las capas de cada uno de los puntos operativos, recuperar, desde el descriptor, valores para los índices de referencia de PTL que correlacionan una capa correspondiente de las capas con una de las estructuras de PTL, para determinar la información de perfil, grada y nivel para la capa correspondiente de las capas; o

45 determinar si el descriptor se incluye en el flujo de bits en función de los tipos de flujo para uno o más flujos elementales de un programa del flujo de bits; o

determinar que el descriptor se incluye en el flujo de bits cuando un tipo de flujo para al menos un flujo elemental de un programa del flujo de bits tiene un valor de 0x27, 0x28, 0x29 o 0x2A.

- 50 4. El procedimiento de la reivindicación 1, en el que:

55 cada una de las estructuras de PTL incluye datos representativos de un valor de espacio\_perfil\_general, un valor general de indicador de grada, una pluralidad de valores de idc\_perfil\_general, un valor de indicador\_compatibilidad\_perfil\_general[i] para el i-ésimo valor de idc\_perfil\_general, un valor de indicador\_origen\_progresivo\_general, un valor de indicador\_origen\_entrelazado\_general, un valor de indicador\_restricción\_no\_empaquetada\_general, un valor de reservado\_cero\_44bits\_general y un valor de idc\_nivel; o

60 el descriptor incluye información para una lista de puntos operativos, incluyendo la información un conjunto de capas de salida asociado para el punto operativo correspondiente, un esquema de partición asociado para el correspondiente punto operativo, una subcapa temporal máxima para el correspondiente punto operativo, una lista de flujos elementales que conforman el punto operativo correspondiente, el número de capas de salida para el punto operativo correspondiente, una correlación de capas contenidas en cada flujo elemental del punto operativo con las estructuras de PTL e información de velocidad de trama para el punto operativo correspondiente.

5. El procedimiento de la reivindicación 1, en el que la extracción de los datos de vídeo (162) comprende:
- 5           determinar un conjunto de capas de salida para uno de los puntos operativos (158) basándose en los datos del descriptor;
- extraer cada capa del conjunto de capas de salida del flujo de bits; y
- extraer del flujo de bits capas de las que dependen las capas del conjunto de capas de salida.
- 10 6. El procedimiento de la reivindicación 5, en el que:
- determinar el conjunto de capas de salida (158) comprende recuperar, desde el descriptor, valores para los elementos sintácticos del conjunto de capas de salida de destino, para cada uno de los puntos operativos, en donde los elementos sintácticos del conjunto de capas de salida de destino especifican los conjuntos de capas de salida de destino asociados a los puntos operativos correspondientes ; o
- 15           determinar el conjunto de capas de salida (158) comprende recuperar, desde el descriptor, los valores de los indicadores para cada una de las capas, en donde los indicadores indican si la capa correspondiente es una capa de salida.
- 20 7. El procedimiento de la reivindicación 1, que comprende además extraer información que indica uno o más flujos elementales de referencia para al menos un flujo elemental del flujo de bits.
- 25 8. El procedimiento de la reivindicación 7, en el que los datos de vídeo de los uno o más flujos elementales de referencia comprenden datos de vídeo en un primer conjunto de una o más capas temporales, y en el que los datos de vídeo del al menos un flujo elemental comprenden datos de vídeo de una capa temporal. capa que es más alta que las capas temporales del primer conjunto.
- 30 9. El procedimiento de la reivindicación 8, que comprende además:
- procesar valores para bits de un elemento sintáctico de bits de dimensión de extensión, en el que al menos un bit del elemento sintáctico de bits de dimensión de extensión tiene un valor que indica que los datos de vídeo del al menos un flujo elemental representan una mejora temporal; o
- 35           procesar un valor para un elemento sintáctico de tipo\_jerarquía del descriptor, en el que el valor indica que un tipo de mejora entre los datos de vídeo del flujo elemental actual y los uno o más flujos elementales de referencia es un tipo auxiliar.
- 40 10. El procedimiento de la reivindicación 1, en el que la extracción del descriptor comprende:
- extraer una pluralidad de descriptores del flujo de bits, siendo cada uno de los descriptores consecutivo en el flujo de bits e incluyendo conjuntos respectivos de estructuras de PTL y datos que asocian cada una de las capas de cada uno de los puntos operativos con una estructura correspondiente de las estructuras de PTL; o
- 45           extraer un grupo de descriptores que siguen inmediatamente a un campo longitud\_información\_programa de una tabla de mapas de programa del flujo de bits; o
- 50           extraer un grupo de descriptores que siguen inmediatamente a un campo longitud\_información\_ES de una tabla de mapas de programa del flujo de bits.
- 55 11. El procedimiento de la reivindicación 1, en el que el descriptor comprende un descriptor de nivel de programa o un descriptor de nivel de flujo elemental.
- 60 12. Un dispositivo (14, 140, 30, 148) para procesar un flujo de bits que incluye datos de vídeo de la codificación de vídeo de alta eficacia, HEVC, comprendiendo el dispositivo:
- una memoria para almacenar datos extraídos del flujo de bits; y
- una o más unidades de procesamiento configuradas para:
- extraer un descriptor del flujo de bits, en donde el flujo de bits incluye capas de datos de vídeo para puntos operativos, independientes del descriptor, de manera que cada punto operativo incluya una o más de las capas de datos de vídeo, en donde, para extraer el descriptor, los uno o más procesadores están configurados para:
- 65

extraer un conjunto de estructuras de perfil, grada y nivel (PTL), y

5 extraer datos que asocian cada una de las capas de cada uno de los puntos operativos con una estructura correspondiente de las estructuras de PTL, en donde los datos que asocian cada una de las capas con la estructura correspondiente de las estructuras de PTL son independientes de las estructuras de PTL en el descriptor;

10 extraer datos de vídeo para uno de los puntos operativos del flujo de bits basándose, al menos en parte, en las estructuras de PTL a las que corresponden las capas de dicho uno de los puntos operativos, y

proporcione los datos de vídeo extraídos a un decodificador de vídeo.

15 **13.** El dispositivo (14, 140, 30, 148) de la reivindicación 12, en el que las una o más unidades de procesamiento están configuradas además para:

20 determinar uno o más perfiles de una norma de codificación de vídeo a la que preste soporte el decodificador de vídeo, una o más gradas de la norma de codificación de vídeo a la que presta soporte el decodificador de vídeo y uno o más niveles dentro de las una o más gradas a las que presta soporte el decodificador de vídeo; y

25 seleccionar dicho uno de los puntos operativos de manera que cada una de las capas de dicho uno de los puntos operativos tenga uno de los perfiles a los que presta soporte el decodificador de vídeo, cada una de las capas de dicho uno de los puntos operativos tenga una de las gradas a las que presta soporte el decodificador de vídeo, y cada una de las capas de dicho uno de los puntos operativos tenga uno de los niveles a los que presta soporte el decodificador de vídeo, como lo indican las estructuras de PTL a las que corresponden las capas de dicho uno de los puntos operativos.

30 **14.** El dispositivo (14, 140, 30, 148) de la reivindicación 12, en el que:

35 las una o más unidades de procesamiento están además configuradas para: iterar por las estructuras de PTL para determinar al menos la información de perfil, la información de grada y la información de nivel para cada una de las estructuras de PTL; y, para cada una de las capas de cada uno de los puntos operativos, recuperar, desde el descriptor, los valores para los índices de referencia de PTL que correlacionan una capa correspondiente de las capas con una de las estructuras de PTL, para determinar la información de perfil, grada y de nivel para la capa correspondiente de las capas; o

40 el descriptor incluye información para una lista de puntos operativos, incluyendo la información un conjunto de capas de salida asociado para el punto operativo correspondiente, un esquema de partición asociado para el correspondiente punto operativo, una subcapa temporal máxima para el correspondiente punto operativo, una lista de flujos elementales que conforman el punto operativo correspondiente, el número de capas de salida para el punto operativo correspondiente, una correlación de capas contenidas en cada flujo elemental del punto operativo con las estructuras de PTL e información de velocidad de trama para el punto operativo correspondiente.

45 **15.** El dispositivo (14, 140, 30, 148) de la reivindicación 12, en el que las una o más unidades de procesamiento están configuradas además para:

50 determinar un conjunto de capas de salida para dicho uno de los puntos operativos, basándose en los datos del descriptor;

extraer cada capa del conjunto de capas de salida del flujo de bits; y

55 extraer capas de las que dependen las capas del conjunto de capas de salida del flujo de bits.

**16.** El dispositivo (14, 140, 30, 148) de la reivindicación 15, en el que, para determinar los conjuntos de capas de salida, los uno o más procesadores están configurados para recuperar, desde el descriptor:

60 los valores para los elementos sintácticos de conjuntos de capas de salida de destino para cada uno de los puntos operativos, en donde los elementos sintácticos de conjuntos de capas de salida de destino especifican los conjuntos de capas de salida de destino asociados a los puntos operativos correspondientes; o

65 valores para los indicadores para cada una de las capas, en donde los indicadores indican si la capa correspondiente es una capa de salida.

17. El dispositivo (14, 140, 30, 148) de la reivindicación 12, en el que las una o más unidades de procesamiento están configuradas además para:

5 extraer información que indique uno o más flujos elementales de referencia para al menos un flujo elemental del flujo de bits; o

10 extraer una pluralidad de descriptores del flujo de bits, siendo cada uno de los descriptores consecutivo en el flujo de bits e incluyendo conjuntos respectivos de estructuras de PTL y datos que asocian cada una de las capas de cada uno de los puntos operativos con una estructura correspondiente de las estructuras de PTL.

18. Un medio de almacenamiento legible por ordenador que tiene almacenadas en él instrucciones que, cuando se ejecutan, hacen que un procesador:

15 extraiga un descriptor de un flujo de bits que incluye datos de vídeo de codificación de vídeo de alta eficacia, HEVC, en donde el flujo de bits incluye capas de datos de vídeo para puntos operativos, independientes del descriptor, de manera que cada punto operativo incluya una o más de las capas de datos de vídeo, en el que las instrucciones que hacen que el procesador extraiga el descriptor comprenden instrucciones que hacen que el procesador:

20 extraiga un conjunto de estructuras de perfil, grada y nivel (PTL); y

25 extraiga los datos que asocian cada una de las capas de cada uno de los puntos operativos con una estructura correspondiente de las estructuras de PTL, y en donde los datos que asocian cada una de las capas con la estructura correspondiente de las estructuras de PTL son independientes de las estructuras de PTL en el descriptor;

30 extraiga datos de vídeo para uno de los puntos operativos del flujo de bits, basándose, al menos en parte, en las estructuras de PTL a las que corresponden las capas de dicho uno de los puntos operativos; y proporcione los datos de vídeo extraídos a un decodificador de vídeo.



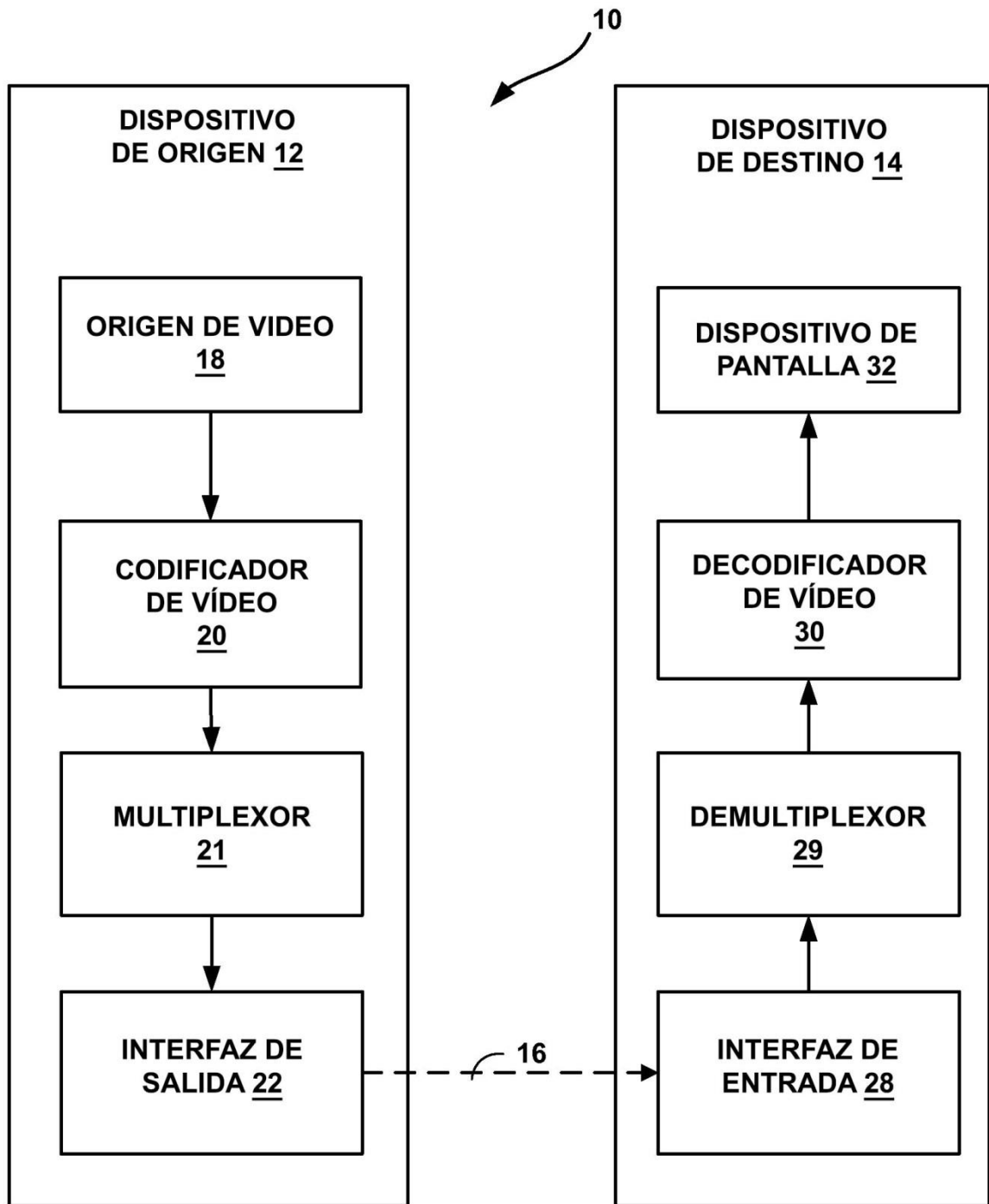


FIG. 1

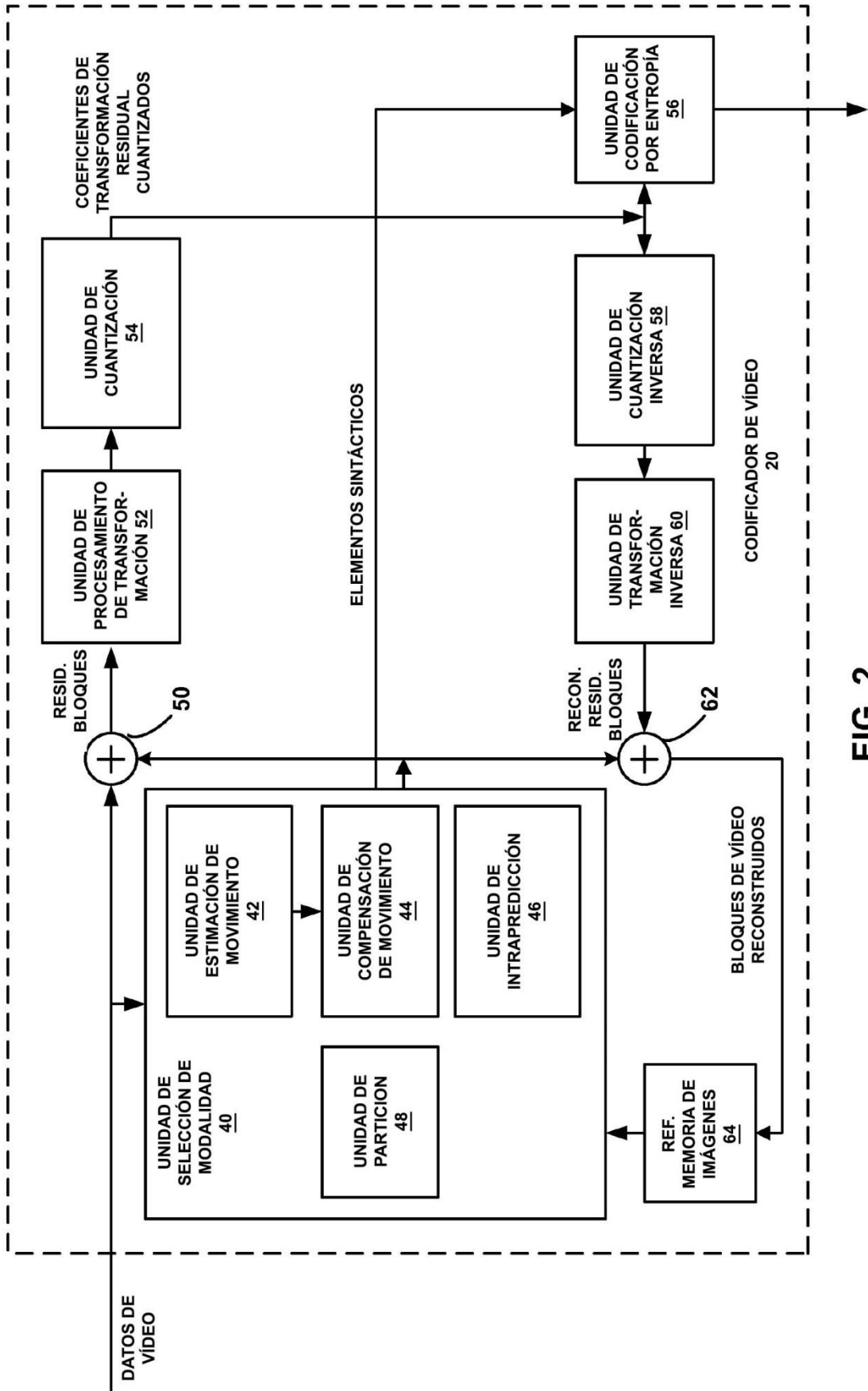


FIG. 2

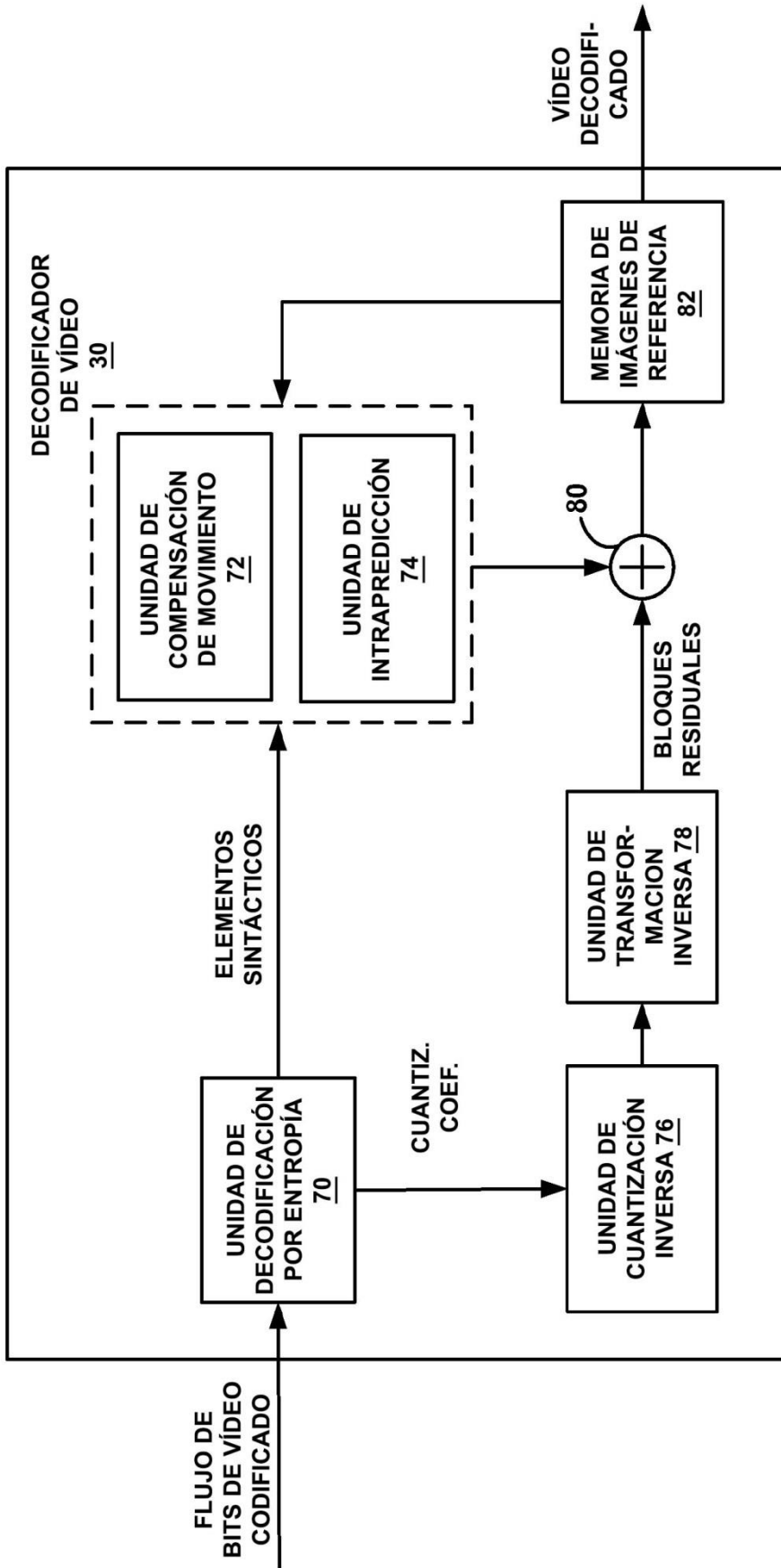


FIG. 3

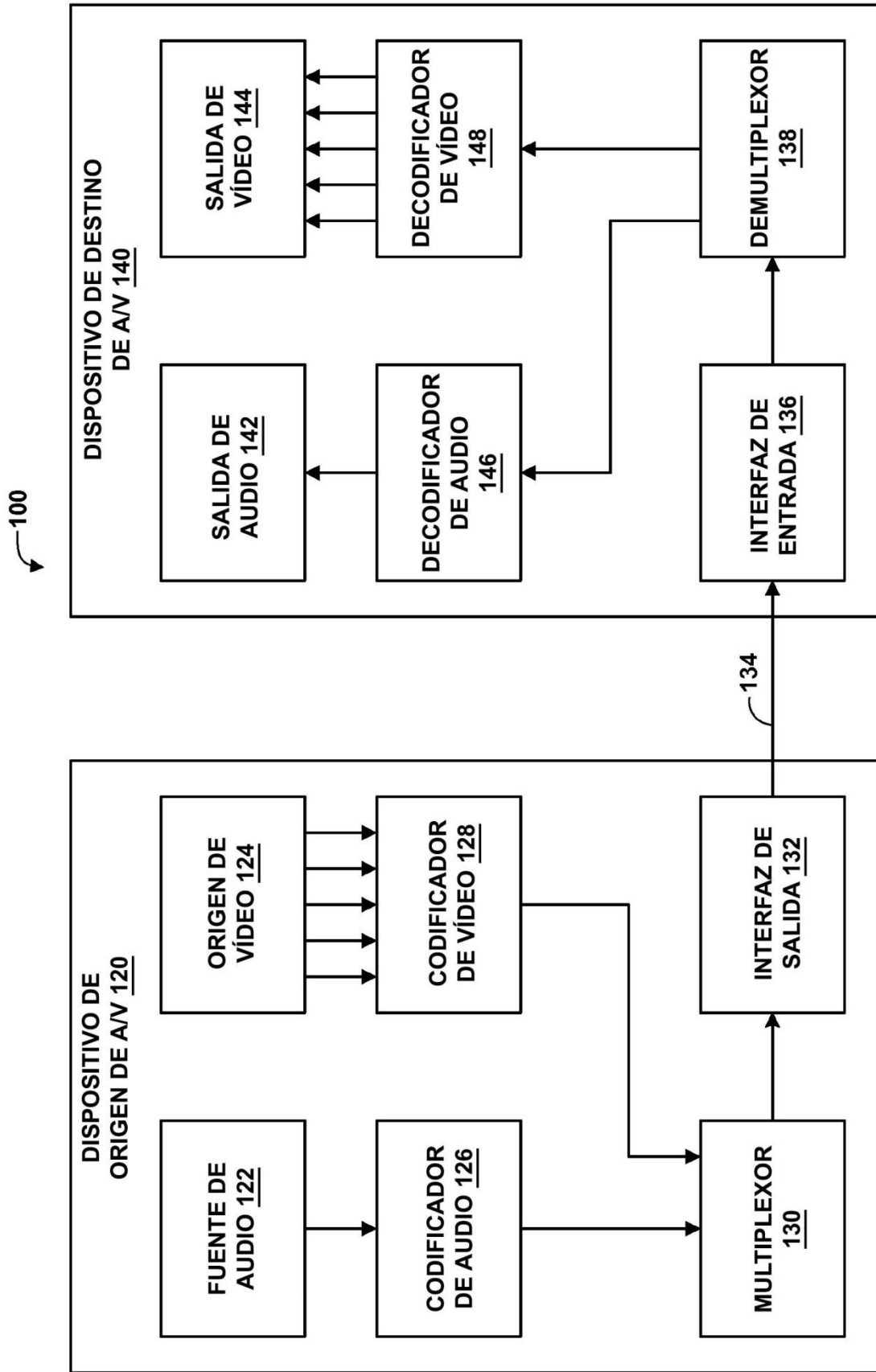
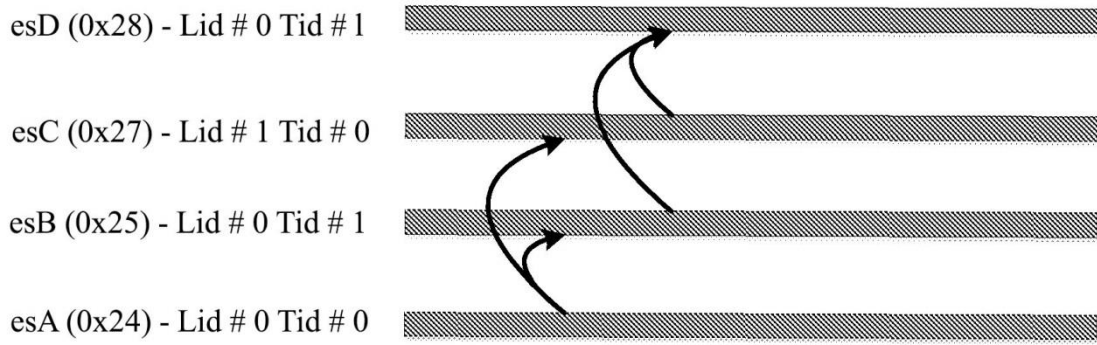
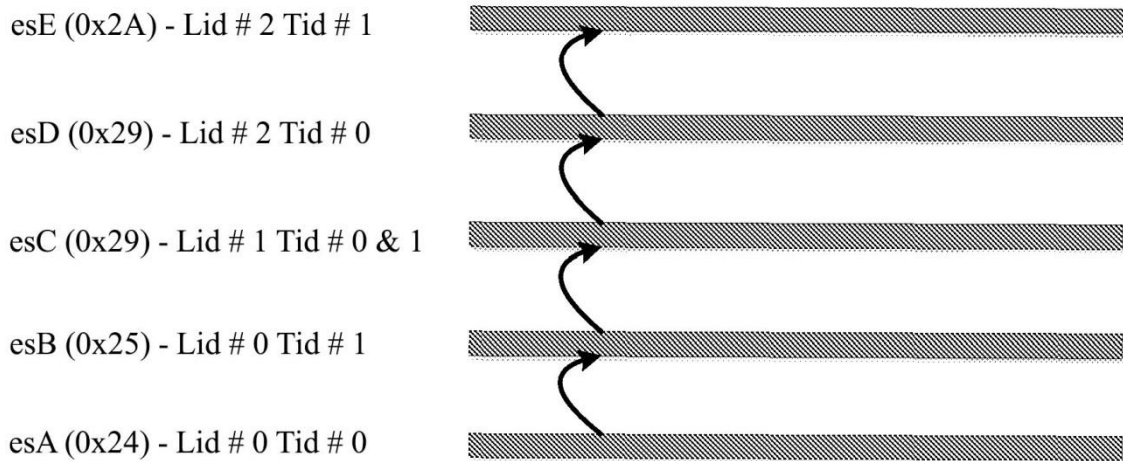


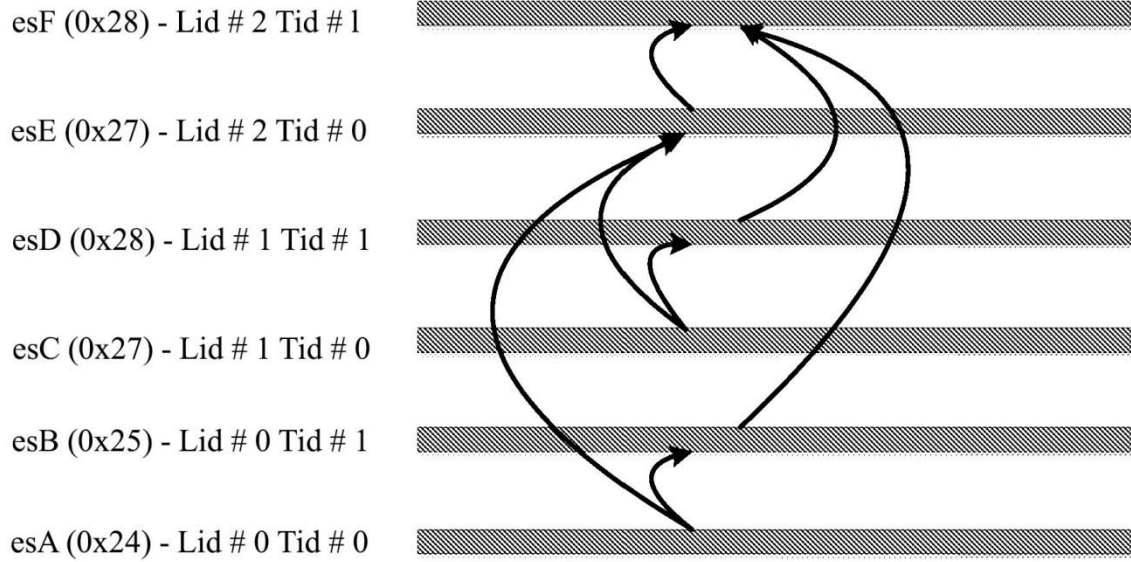
FIG. 4



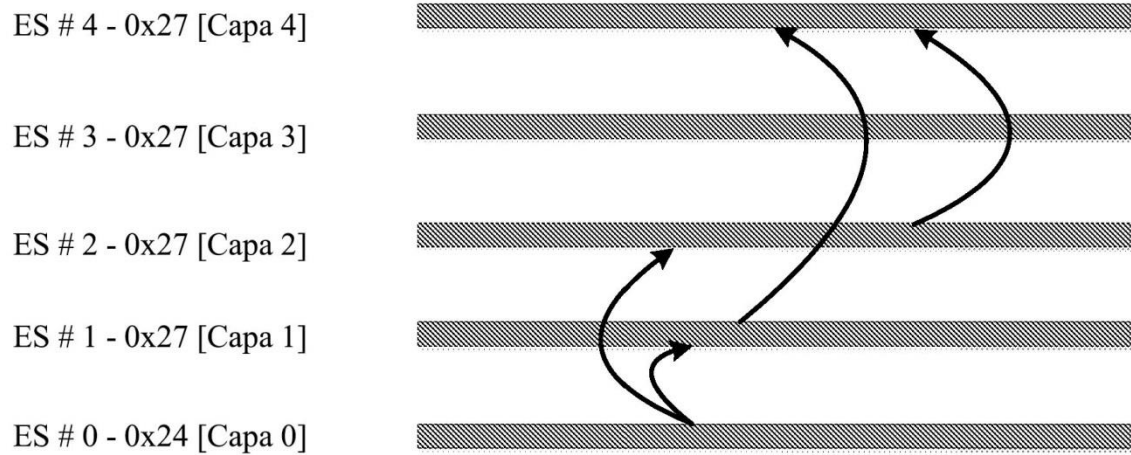
**FIG. 5**



**FIG. 6**



**FIG. 7**



**FIG. 8**

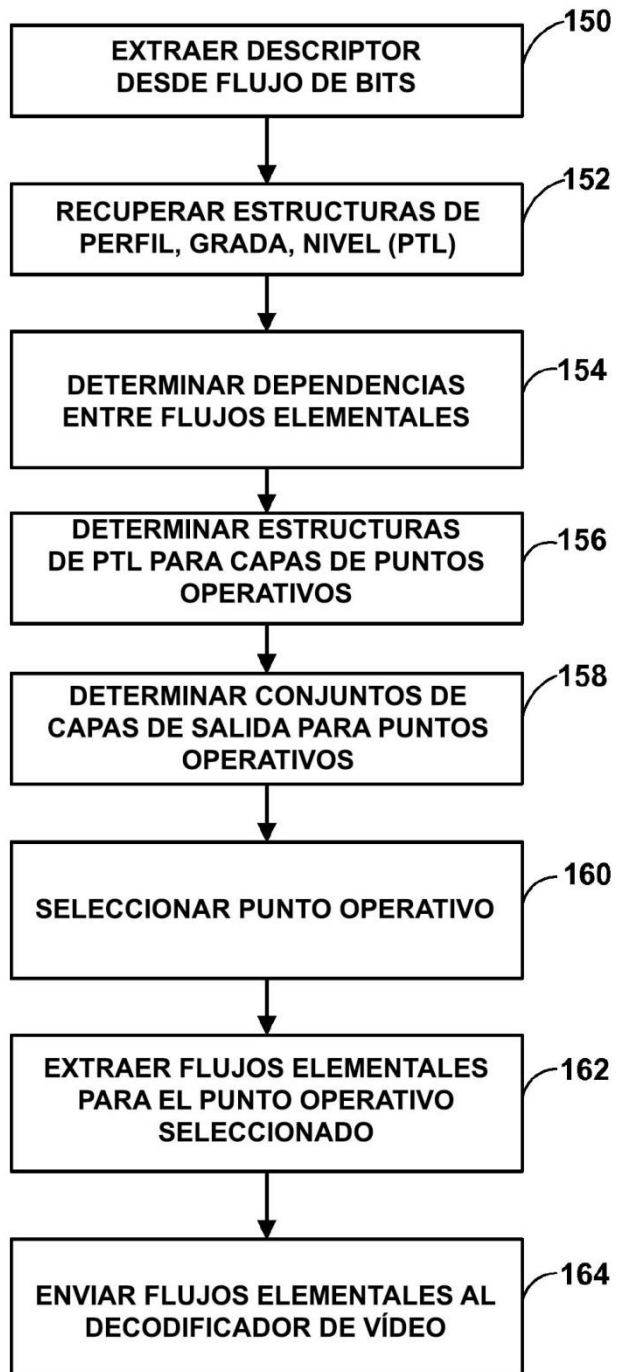


FIG. 9