(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 902 675**

(51) Int. CI.:

H04N 19/70 (2014.01)
H04N 21/2343 (2011.01)
H04N 21/845 (2011.01)
H04N 21/854 (2011.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **10.02.2016 PCT/US2016/017280**

(87) Fecha y número de publicación internacional: **18.08.2016 WO16130635**

(96) Fecha de presentación y número de la solicitud europea: **10.02.2016 E 16708519 (0)**

(97) Fecha y número de publicación de la concesión europea: **08.12.2021 EP 3257250**

(54) Título: **Diseño de señalización de entrada de muestra y de punto de operación en un formato de archivo de vídeo estratificado**

(30) Prioridad:

11.02.2015 US 201562115075 P
09.02.2016 US 201615019634

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
29.03.2022

(73) Titular/es:

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

(72) Inventor/es:

HENDRY, FNU y
WANG, YE-KUI

(74) Agente/Representante:

ISERN JARA, Jorge

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

Diseño de señalización de entrada de muestra y de punto de operación en un formato de archivo de vídeo estratificado

Campo técnico

Esta divulgación se refiere a la codificación de vídeo.

Antecedentes

Las capacidades de vídeo digital pueden ser incorporadas dentro de una amplia gama de dispositivos, incluidos televisores digitales, sistemas de radiodifusión directa digital, sistemas de radiodifusión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, ordenadores de tableta, lectores de libros electrónicos, cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos móviles o de radio por satélite, así denominados "teléfonos inteligentes", dispositivos de videoteleconferencia, dispositivos de transmisión de vídeo, y similares. Los dispositivos de vídeo digital implementan técnicas de compresión de vídeo, tales como aquellas descritas en los estándares definidos por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Parte 10, codificación de vídeo avanzada (AVC), el estándar de codificación de vídeo de alta eficiencia (HEVC) actualmente en desarrollo, y extensiones de tales estándares. Los dispositivos de vídeo pueden transmitir, recibir, codificar, decodificar, y/o almacenar información de vídeo digital más eficientemente implementando tales técnicas de compresión de vídeo.

Las técnicas de compresión de vídeo realizan predicción espacial (intraimagen) y/o predicción temporal (interimagen) para reducir o eliminar la redundancia inherente en las secuencias de vídeo. Para la codificación de vídeo basada en bloques, un segmento de vídeo (esto es, un fotograma de vídeo o una porción de un fotograma de vídeo) puede ser particionado en bloques de vídeo, a los cuales también se puede hacer referencia como bloques de árbol, unidades de codificación (CU) y/o nodos de codificación. Los bloques de vídeo en un segmento intracodificado (I) de una imagen son codificados usando predicción espacial con respecto a muestras de referencia en bloques colindantes en la misma imagen. Los bloques de vídeo en un segmento intercodificado (P o B) de una imagen pueden usar predicción espacial con respecto a muestras de referencia en bloques colindantes en la misma imagen o predicción temporal con respecto a muestras de referencia en otras imágenes de referencia. A las imágenes se puede hacer referencia como fotogramas, y a las imágenes de referencia se puede hacer referencia como fotogramas de referencia.

Después de que los datos de vídeo han sido codificados, los datos de vídeo pueden ser paquetizados para la transmisión o el almacenamiento. Los datos de vídeo pueden ser reunidos en un archivo de vídeo conforme a cualquiera de una variedad de estándares, tal como el formato de archivo multimedia base de la Organización Internacional de Normalización (ISO) y extensiones de este, tal como AVC.

El documento US 2012/013746 describe técnicas para multiplexar componentes de vídeo.

El documento US 2013/287366 describe técnicas para identificar conjuntos de parámetros en archivos de vídeo.

JONATAN SAMUELSSON ET AL, "HEVC file format parallelism indication" [Indicación de paralelismo de formato de archivo HEVC], ENCUENTRO 102. MPEG; 15-10-2012 - 19-10-2012; SHANGHAI; (MOTION PICTURE EXPERT GROUP [Grupo experto en imagen en movimiento] o ISO/IEC JTC1/SC29/WG11), (20121018), n.º m26877, propone añadir dos parámetros al registro de configuración de decodificador HEVC.

"Text of ISO/IEC 14496-15:2010 PDAM 2 Carriage of High efficiency Video Coding (HEVC)" [Texto de ISO/IEC 14496-15:2010 PDAM 2 Transporte de codificación de vídeo de alta eficiencia (HEVC)], ENCUENTRO 100. MPEG; 30-4-2012 - 4-5-2012; GÉNOVA; (MOTION PICTURE EXPERT GROUP [Grupo experto en imagen en movimiento] o ISO/IEC JTC1/SC29/WG11), (20120508), n.º N12647, define un formato de almacenamiento basado en y compatible con el Formato de Archivo Multimedia Base ISO.

"WD of ISO/IEC 14496-15:2013 AMD 1 Enhanced carriage of HEVC and support of MVC with depth information" [Borrador de trabajo de ISO/IEC 14496-15:2013 AMD 1 Transporte mejorado de HEVC y soporte de MVC con información de profundidad], 108, ENCUENTRO MPEG; 31/3/2014 - 4/4/2014; Valencia, (Motion Picture Expert Group [Grupo experto en imagen en movimiento] o ISO/IEC JTC1/SC29/WG11) n.º N14328, 16 de mayo de 2014 divulga señalización de perfil, nivel y capa capturada por la caja oinf.

El documento US 2012/023250 A1 divulga que las características de los puntos de operación pueden incluir un indicador de perfil, un indicador de nivel, una tasa de fotogramas para el punto de operación, una tasa de bits promedio para el punto de operación, una tasa de bits máxima para el punto de operación, una resolución espacial para el punto de operación, un número de vistas a transmitir para el punto de operación, y/o un número de vistas a ser decodificadas para el punto de operación.

VINOD KUMAR MALAMAL VADAKITAL et al: “Proposals and Comments about the Specification for Layers-HEVC in 14496-15” [Propuestas y comentarios sobre la especificación para HEVC de capas en 14496-15”, encuentro 109. MPEG; 7-7-2014 0 11-7-2014; Sapporo; (MOTION PICTURE EXPERT GROUP [Grupo experto en imagen en movimiento] o ISO/IEC JTC1/SC29/WG11), n.º m34125, 9 de julio de 2014 (2014-07-09),
se refiere a una OperationPointsInformationBox ‘oinf’. La caja ‘oinf’ proporciona información sobre todos los puntos de operación definidos en el flujo de bits L-HEVC codificado, sus capas componentes, las limitaciones de perfil-nivel-capa de cada punto de operación, las características de capas en un punto de operación, las relaciones de dependencia entre capas transportadas en el flujo de bits, y los tipos de escalabilidad definidos en el flujo de bits.

Sumario

En general, esta divulgación se refiere al almacenamiento de contenido de vídeo en un archivo. En algunos ejemplos, las técnicas de la divulgación están basadas en el formato de archivo multimedia base (ISOBMFF) de la Organización Internacional de Normalización (ISO). Algunos ejemplos de la esta divulgación se refieren a métodos para el almacenamiento de flujos de vídeo que contienen múltiples capas codificadas, donde cada capa puede ser una capa escalable, una vista de textura, una vista de profundidad, etc., y los métodos pueden aplicarse al almacenamiento de codificación de vídeo de alta eficiencia multivista (MV-HEVC), HEVC escalable (SHVC), HEVC tridimensional (3D-HEVC), y otros tipos de datos de vídeo. La invención está definida en las reivindicaciones independientes adjuntas. Características opcionales están expuestas en las reivindicaciones dependientes.

En un ejemplo, un método de procesar datos de vídeo multicapa incluye obtener los datos de vídeo multicapa; almacenar los datos de vídeo multicapa en un formato de archivo; almacenar información de formato de representación para cada punto de operación de los datos de vídeo multicapa en una caja de información de puntos de operación (oinf) para el formato de archivo; y generar un archivo de datos de vídeo formateado de acuerdo con el formato de archivo.

En otro ejemplo, un método de procesar datos de vídeo multicapa incluye obtener un archivo de datos de vídeo multicapa formateado de acuerdo con un formato de archivo; para el formato de archivo, determinar la información de formato de representación para cada punto de operación de los datos de vídeo multicapa en una caja de información de puntos de operación (oinf) para el formato de archivo; y decodificar los datos de vídeo multicapa basándose en la información de formato de representación determinada.

En otro ejemplo, un dispositivo de vídeo para procesar datos de vídeo multicapa incluye un medio de almacenamiento de datos configurado para almacenar los datos de vídeo multicapa y uno o más procesadores configurados para: obtener los datos de vídeo multicapa; almacenar los datos de vídeo multicapa en un formato de vídeo; almacenar información de formato de representación para cada punto de operación de los datos de vídeo multicapa en una caja de información de puntos de operación (oinf) para el formato de archivo; y generar un archivo de datos de vídeo formateado de acuerdo con el formato de archivo.

En otro ejemplo, un dispositivo de vídeo para procesar datos de vídeo multicapa incluye un medio de almacenamiento de datos configurado para almacenar los datos de vídeo multicapa y uno o más procesadores configurados para obtener un archivo de datos de vídeo multicapa formateado de acuerdo con un formato de archivo; para el formato de archivo, determinar información de formato de representación para cada punto de operación de los datos de vídeo multicapa en una caja de información de puntos de operación (oinf) para el formato de archivo; y decodificar los datos de vídeo multicapa basándose en la información de formato de representación determinada.

En otro ejemplo, un dispositivo de vídeo para procesar datos de vídeo multicapa incluye medios para obtener los datos de vídeo multicapa; medios para almacenar los datos de vídeo multicapa en un formato de archivo; medios para almacenar información de formato de representación para cada punto de operación de los datos de vídeo multicapa en una caja de información de puntos de operación (oinf) para el formato de archivo; y medios para generar un archivo de datos de vídeo formateado de acuerdo con el formato de archivo.

En otro ejemplo, un medio de almacenamiento legible por ordenador almacena instrucciones que cuando son ejecutadas hacen que uno o más procesadores obtengan datos de vídeo multicapa; almacenen los datos de vídeo multicapa en un formato de archivo; almacenen información de formato de representación para cada punto de operación de los datos de vídeo multicapa en una caja de información de puntos de operación (oinf) para el formato de archivo; y generen un archivo de datos de vídeo formateado de acuerdo con el formato de archivo.

Los detalles de uno o más ejemplos de la divulgación están expuestos en los dibujos adjuntos y la descripción de abajo. Otras características, objetos, y ventajas serán aparentes a partir de la descripción, los dibujos, y las reivindicaciones.

Descripción breve de los dibujos

La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación y decodificación de vídeo de ejemplo que puede usar las técnicas descritas en esta divulgación.

La FIG. 2 es un diagrama de bloques que ilustra un codificador de vídeo de ejemplo que puede implementar las técnicas descritas en esta divulgación.
La FIG. 3 es un diagrama de bloques que ilustra un decodificador de vídeo de ejemplo que puede implementar las técnicas descritas en esta divulgación.
La FIG. 4 es un diagrama de bloques que ilustra un conjunto de dispositivos de ejemplo que forman parte de una red.
La FIG. 5A es un diagrama conceptual que ilustra una estructura de ejemplo de un archivo de conformidad con una o más técnicas de esta divulgación.
La FIG. 5B es un diagrama conceptual que ilustra una estructura de ejemplo de un archivo de conformidad con una o más técnicas de esta divulgación.
La FIG. 6 es un diagrama conceptual que ilustra una estructura de ejemplo de un archivo de conformidad con una o más técnicas de esta divulgación.
La FIG. 7 es un diagrama de flujo que ilustra una operación de ejemplo de un dispositivo de generación de archivos de conformidad con una o más técnicas de esta divulgación.
La FIG. 8 es un diagrama de flujo que ilustra una operación de ejemplo de un dispositivo de lectura de archivos de conformidad con una o más técnicas de esta divulgación.

Descripción detallada

El formato de archivo multimedia base ISO (ISOBMFF) es un formato de archivo para almacenar datos de medios. El ISOBMFF es ampliable para soportar el almacenamiento de datos de vídeo conforme a estándares de codificación de vídeo particulares. Por ejemplo, el ISOBMFF ha sido previamente ampliado para soportar el almacenamiento de datos de vídeo conforme a los estándares de codificación de vídeo H.264/AVC y codificación de vídeo de alta eficiencia (HEVC). Además, el ISOBMFF ha sido previamente ampliado para soportar el almacenamiento de datos de vídeo conforme a las extensiones de codificación multivista (MVC) y codificación de vídeo escalable (SVC) de H.264/AVC. MV-HEVC, 3D-HEVC, y SHVC son extensiones del estándar de codificación de vídeo HEVC que soportan datos de vídeo multicapa. Las características añadidas al ISOBMFF para el almacenamiento de datos de vídeo conforme a las extensiones MVC y SVC de H.264/AVC no son suficientes para el almacenamiento efectivo de datos de vídeo conforme a MV-HEVC, 3D-HEVC, y SHVC. En otras palabras, varios problemas pueden surgir si se intentara usar la extensión del ISOBMFF para el almacenamiento de datos de vídeo conforme a las extensiones MVC y SVC de H.264/AVC para el almacenamiento de datos de vídeo conforme a MV-HEVC, 3D-HEVC, y SHVC.

Por ejemplo, a diferencia de un flujo de bits que se adecua a las extensiones MVC o SVC de H.264/AVC, un flujo de bits que se adecua a MV-HEVC, 3D-HEVC, o SHVC puede incluir unidades de acceso que contienen imágenes de punto de acceso intraaleatorio (IRAP) e imágenes no IRAP. Una unidad de acceso que contiene imágenes IRAP e imágenes no IRAP puede ser usada para el acceso aleatorio en MV-HEVC, 3D-HEVC, y SHVC. Sin embargo, el ISOBMFF y las extensiones existentes de este no proporcionan una forma de identificar tales unidades de acceso. Esto puede dificultar la capacidad de un dispositivo de computación para realizar acceso aleatorio, conmutación de capa, y otras tales funciones asociadas con datos de vídeo multicapa.

Aunque gran parte de la descripción de las técnicas de esta divulgación describe MV-HEVC, 3D-HEVC, y SHVC, el lector apreciará que las técnicas de esta divulgación pueden ser aplicables a otros estándares de codificación de vídeo y/o extensiones de estos.

Como será explicado en más detalle abajo, el archivo que se adecua al formato de archivo HEVC puede incluir una serie de objetos, llamados cajas. Una caja puede ser un bloque de construcción orientado al objeto definido por un identificador de tipo único y una longitud. Esta divulgación describe técnicas relativas a generar archivos de acuerdo con un formato de archivo y, más particularmente, describe técnicas para localizar ciertos tipos de información en ciertas cajas para mejorar potencialmente la capacidad de un dispositivo de reproducción para procesar archivos que incluyen múltiples puntos de operación.

Aunque gran parte de la descripción de las técnicas de esta divulgación describe MV-HEVC, 3D-HEVC, y SHVC, el lector apreciará que las técnicas de esta divulgación pueden ser aplicables a otros estándares de codificación de vídeo y/o extensiones de estos.

La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación y decodificación de vídeo 10 de ejemplo que puede usar las técnicas descritas en esta divulgación. Como está mostrado en la FIG. 1, el sistema 10 incluye un dispositivo de origen 12 que genera datos de vídeo codificados a ser decodificados en un momento posterior por un dispositivo de destino 14. El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender cualquiera de una amplia gama de dispositivos, incluidos ordenadores de escritorio, ordenadores notebook (esto es, portátiles), ordenadores de tableta, decodificadores, terminales telefónicos tales como los así denominados teléfonos “inteligentes”, así denominadas almohadillas “inteligentes”, televisores, cámaras, dispositivos de pantalla, reproductores de medios digitales, consolas de videojuegos, dispositivo de transmisión 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. El dispositivo de origen 12 y el dispositivo de destino 14 pueden ser considerados dispositivos de vídeo.

En el ejemplo de la FIG. 1, el dispositivo de origen 12 incluye una fuente de vídeo 18, codificador de vídeo 20 y una interfaz de salida 22. En algunos casos, la interfaz de salida 22 puede incluir un modulador/demodulador (módem) y/o un transmisor. En el dispositivo de origen 12, la fuente de vídeo 18 puede incluir una fuente tal como un dispositivo de captura de vídeo, p. ej., una cámara de vídeo, un archivo de vídeo que contiene vídeo capturado previamente, una interfaz de transmisión de vídeo para recibir vídeo desde un proveedor de contenido de vídeo, y/o un sistema de gráficos computacionales para generar datos de gráficos computacionales como el vídeo de origen, o una combinación de tales fuentes. Sin embargo, las técnicas descritas en esta divulgación pueden ser aplicables a la codificación de vídeo en general, y pueden ser aplicadas a aplicaciones por cable y/o inalámbricas.

El codificador de vídeo 20 puede codificar el vídeo capturado, precapturado, o generado por ordenador. El dispositivo de origen 12 puede transmitir los datos de vídeo codificados directamente al dispositivo de destino 14 a través de la interfaz de salida 22 del dispositivo de origen 12. Los datos de vídeo codificados pueden también (o alternativamente) ser almacenados en el dispositivo de almacenamiento 33 para el acceso posterior por parte del dispositivo de destino 14 u otros dispositivos, para la decodificación y/o reproducción.

El dispositivo de destino 14 incluye una interfaz de entrada 28, un decodificador de vídeo 30, y un dispositivo de pantalla 32. En algunos casos, la interfaz de entrada 28 puede incluir un receptor y/o un módem. La interfaz de entrada 28 del dispositivo de destino 14 recibe los datos de vídeo codificados a través del enlace 16. Los datos de vídeo codificados comunicados a través del enlace 16, o proporcionados en el dispositivo de almacenamiento 33, pueden incluir una variedad de elementos de sintaxis generados por el codificador de vídeo 20 para el uso por parte de un decodificador de vídeo, tal como el decodificador de vídeo 30, en la decodificación de los datos de vídeo. Tales elementos de sintaxis pueden estar incluidos con los datos de vídeo codificados transmitidos en un medio de comunicación, almacenados en un medio de almacenamiento, o almacenados en un servidor de archivos.

El dispositivo de pantalla 32 puede estar integrado con, o puede ser externo a, el dispositivo de destino 14. En algunos ejemplos, el dispositivo de destino 14 puede incluir un dispositivo de pantalla integrado y también puede estar configurado para interactuar con un dispositivo de pantalla externo. En otros ejemplos, el dispositivo de destino 14 puede ser un dispositivo de pantalla. En general, el dispositivo de pantalla 32 muestra los datos de vídeo decodificados a un usuario, y puede comprender cualquiera de una variedad de dispositivos de pantalla tales como una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodo orgánico de emisión de luz (OLED), u otro tipo de dispositivo de pantalla.

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden ser cada uno implementados como cualquiera de una variedad de circuitería de codificador apropiada, tal como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados de aplicación específica (ASIC), matrices de puertas programables en campo (FPGA), lógica discreta, software, hardware, firmware o cualquier combinación de estos. Cuando las técnicas son implementadas parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio legible por ordenador no transitorio apropiado y ejecutar las instrucciones en hardware usando uno o más procesadores para llevar a cabo las técnicas de esta divulgación. Cada uno del codificador de vídeo 20 y decodificador de vídeo 30 puede estar incluido en uno o más codificadores o decodificadores, cualquiera de los cuales puede estar integrado como parte de un codificador/decodificador (CÓDEC) combinado en un dispositivo respectivo.

El dispositivo de destino 14 puede recibir los datos de vídeo codificados a ser decodificados a través de un enlace 16. El enlace 16 puede comprender cualquier tipo de medio o dispositivo capaz de mover los datos de vídeo codificados desde el dispositivo de origen 12 al dispositivo de destino 14. En un ejemplo, el enlace 16 puede comprender un medio de comunicación para habilitar el dispositivo de origen 12 para transmitir datos de vídeo codificados directamente al dispositivo de destino 14 en tiempo real. Los datos de vídeo codificados pueden ser modulados de acuerdo con un estándar de comunicación, tal como un protocolo de comunicación inalámbrica, y transmitidos al dispositivo de destino 14. El medio de comunicación puede comprender cualquier medio de comunicación alámbrico o inalámbrico, tal como un espectro de radiofrecuencia (RF) o una o más líneas de transmisión físicas. El medio de comunicación puede formar parte de una red basada en paquetes, tal como una red de área local, una red de área extendida, o una red global tal como la Internet. El medio de comunicación puede incluir rúters, interruptores, estaciones base, o cualquier otro equipo que puede ser útil para facilitar la comunicación desde el dispositivo de origen 12 al dispositivo de destino 14.

Alternativamente, la interfaz de salida 22 puede transmitir datos codificados a un dispositivo de almacenamiento 33. Similarmente, la interfaz de entrada 28 puede acceder al dispositivo de almacenamiento 33 de datos codificados. El dispositivo de almacenamiento 33 puede incluir cualquiera de una variedad de medios de almacenamiento de datos accedidos localmente o distribuidos tales como un disco duro, discos Blu-Ray, DVD, CD-ROM, memoria flash, memoria volátil o no volátil, o cualquier otro medio de almacenamiento digital apropiado para almacenar datos de vídeo codificados. En un otro ejemplo, el dispositivo de almacenamiento 33 puede corresponder a un servidor de archivos u otro dispositivo de almacenamiento intermedio que puede guardar el vídeo codificado generado por el dispositivo de origen 12. El dispositivo de destino 14 puede acceder a los datos de vídeo almacenados desde el dispositivo de almacenamiento 33 por streaming o descarga. El servidor de archivos 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. Servidores de archivos de ejemplo incluyen un servidor web (p. ej., para un sitio web), un servidor FTP, dispositivos

de almacenamiento conectados a la red (NAS), o un disco duro 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 (p. ej., una conexión Wi-Fi), una conexión por cable (p. ej., DSL, módem de cable, etc.), o una combinación de ambos que es apropiada para acceder a datos de vídeo codificados almacenados en un servidor de archivos. La transmisión de datos de vídeo codificados desde el dispositivo de almacenamiento 33 puede ser una transmisión por streaming, una transmisión por descarga, o una combinación de ambas.

Las técnicas de esta divulgación no están necesariamente limitadas a aplicaciones o contextos inalámbricos. Las técnicas pueden ser aplicadas a codificación de vídeo en soporte de cualquiera de una variedad de aplicaciones multimedia, tales como radiotransmisiones televisivas por aire, transmisiones televisivas por cable, transmisiones televisivas por satélite, transmisiones de vídeo por streaming, p. ej., a través de Internet, codificación de vídeo digital para el almacenamiento 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 estar configurado para soportar la transmisión de vídeo unidireccional o bidireccional para soportar aplicaciones tales como transmisión de vídeo, reproducción de vídeo, radiotransmisión de vídeo, y/o videotelefonía.

Además, en el ejemplo de la FIG. 1, el sistema de codificación de vídeo 10 puede incluir un dispositivo de generación de archivos 34. El dispositivo de generación de archivos 34 puede recibir datos de vídeo codificados generados por el dispositivo de origen 12 y generar un archivo que incluye los datos de vídeo codificados. El dispositivo de destino 14 puede recibir, o bien directamente o bien a través del dispositivo de almacenamiento 33, el archivo generado por el dispositivo de generación de archivos 34. En varios ejemplos, el dispositivo de generación de archivos 34 puede incluir varios tipos de dispositivos de computación. Por ejemplo, el dispositivo de generación de archivos 34 puede comprender un Media Aware Network Element (MANE), un dispositivo de computación de servidor, un dispositivo de computación personal, un dispositivo de computación de propósito específico, un dispositivo de computación comercial, u otro tipo de dispositivo de computación. En algunos ejemplos, el dispositivo de generación de archivos 34 es parte de una red de entrega de contenidos. El dispositivo de generación de archivos 34 puede recibir los datos de vídeo codificados desde el dispositivo de origen 12 a través de un canal tal como el enlace 16. Además, el dispositivo de destino 14 puede recibir el archivo desde el dispositivo de generación de archivos 34 a través de un canal tal como el enlace 16.

En algunas configuraciones, el dispositivo de generación de archivos 34 puede ser un dispositivo de vídeo separado del dispositivo de origen 12 y el dispositivo de destino 14, mientras que en otras configuraciones, el dispositivo de generación de archivos 34 puede ser implementado como un componente del dispositivo de origen 12 o dispositivo de destino 14. En implementaciones donde el dispositivo de generación de archivos 34 es un componente del dispositivo de origen 12 o dispositivo de destino 14, entonces el dispositivo de generación de archivos 34 puede compartir algunos de los mismos recursos, tales como memorias, procesadores, y otro hardware, utilizados por el codificador de vídeo 20 y el decodificador de vídeo 30. En implementaciones donde el dispositivo de generación de archivos 34 es un dispositivo separado, entonces el dispositivo de generación de archivos puede incluir su propia memoria, procesadores, y otras unidades de hardware.

En otros ejemplos, el dispositivo de origen 12 u otro dispositivo de computación puede generar un archivo que incluye los datos de vídeo codificados. Sin embargo, para facilidad de explicación, esta divulgación describe el dispositivo de generación de archivos 34 como que genera el archivo. No obstante, se debe entender que tales descripciones son aplicables a los dispositivos de computación en general.

El codificador de vídeo 20 y el decodificador de vídeo 30 pueden operar de acuerdo con un estándar de compresión de vídeo, tal como el estándar de codificación de vídeo de alta eficiencia (HEVC) o una extensión de este. Al estándar HEVC también se puede hacer referencia como ISO/IEC 23008-2. Recientemente, el diseño de HEVC ha sido finalizado por el Equipo de colaboración conjunta sobre codificación de vídeo (JCT-VC) del grupo de expertos en codificación de vídeo ITU-T (VCEG) y el grupo de expertos en imagen en movimiento ISO/IEC (MPEG). La última especificación del borrador HEVC, y referido como HEVC WD en adelante, está disponible desde http://phenix.int-evry.fr/jct/doc_end_user/documents/14_Vienna/wg11/JCTVC-N1003-v1.zip. La extensión multivista para HEVC, a saber MV-HEVC, también está siendo desarrollada por el JCT-3V. Un borrador de trabajo (WD) reciente de MV-HEVC, titulado “MV-HEVC Draft Text 5” [Texto borrador MV-HEVC 5] y al que se hace referencia como MV-HEVC WD5 en adelante, está disponible desde http://phenix.it-sudpar-is.eu/jct2/doc_end_user/documents/5_Vienna/wg11/JCT3V-E1004-v6.zip. La extensión escalable para HEVC, llamada SHVC, también está siendo desarrollada por el JCT-VC. Un borrador de trabajo (WD) reciente de SHVC, titulado “High efficiency video coding (HEVC) scalable extension draft 3” [Borrador de extensión escalable de high efficiency video coding (HEVC) 3] y al que se hace referencia como SHVC WD3 en adelante, está disponible desde http://phenix.it-sudpar-is.eu/jct/doc_end_user/documents/14_Vienna/wg11/JCTVC-N1008-v3.zip. Un borrador de trabajo (WD) reciente de la extensión de gama de HEVC, está disponible desde http://phenix.int-evry.fr/jct/doc_end_user/docu-ments/14_Vienna/wg11/JCTVC-N1005-v3.zip. Un borrador de trabajo (WD) reciente de la extensión 3D de HEVC, a saber 3D-HEVC, titulado “3D-HEVC Draft Text 1” [Texto borrador 3D-HEVC 1] está disponible desde http://phenix.int-evry.fr/jt2/doc_end_user/docu-ments/5_Vienna/wg11/JCT3V-E1001-v3. El codificador de vídeo 20 y el decodificador de vídeo 30 pueden operar de acuerdo con uno o más de estos estándares.

Alternativamente, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden operar de acuerdo con otros estándares industriales o propietarios, tal como el estándar ITU-T H.264, al que se hace referencia alternativamente como MPEG-4, Parte 10, codificación de vídeo avanzada (AVC), o extensiones de tales estándares. Las técnicas de esta divulgación, sin embargo, no están limitadas a ningún estándar de codificación particular. Otros ejemplos de estándares de compresión de vídeo incluyen ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 o ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual e ITU-T H.264 (también conocido como ISO/IEC MPEG-4 AVC), incluidas sus extensiones de codificación de vídeo escalable (SVC) y codificación de vídeo multivista (MVC).

Aunque no está mostrado en la FIG. 1, en algunos aspectos, el codificador de vídeo 20 y el decodificador de vídeo 30 pueden estar cada uno integrados con un codificador y decodificador de audio, y pueden incluir unidades MUX-DEMUX apropiadas, u otro hardware y software, para gestionar la codificación tanto de audio como vídeo en un flujo de datos común o flujos de datos separados. Si es aplicable, en algunos ejemplos, las unidades MUX-DEMUX pueden adecuarse al protocolo de multiplexor ITU H.223, u otros protocolos tales como el protocolo de datagramas de usuario (UDP).

El JCT-VC desarrolló el estándar HEVC. Los esfuerzos de estandarización de HEVC están basados en un modelo en desarrollo de un dispositivo de codificación de vídeo al que se hace referencia como el Modelo de Prueba HEVC (HM). El HM presume varias capacidades adicionales de los dispositivos de codificación de vídeo en relación con los dispositivos existentes de acuerdo con, p. ej., ITU-T H.264/AVC. Por ejemplo, mientras que el H.264/AVC proporciona nueve modos de codificación de intrapredicción, el HM puede proporcionar tantos como treinta y tres modos de codificación de intrapredicción.

En general, el modelo de trabajo del HM describe que un fotograma o imagen de vídeo puede ser dividido en una secuencia de bloques de árbol o unidades de codificación máxima (LCU) que incluyen muestras tanto luma como croma. A los bloques de árbol también se puede hacer referencia como unidades de árbol de codificación (CTU). Un bloque de árbol tiene un propósito similar como un macrobloque del estándar H.264/AVC. Un segmento incluye un número de bloques de árbol consecutivos en orden de codificación. Un fotograma o imagen de vídeo puede ser particionado en uno o más segmentos. Cada bloque de árbol puede ser dividido en unidades de codificación (CU) de acuerdo con un árbol cuaternario. Por ejemplo, un bloque de árbol, como un nodo de raíz del árbol cuaternario, puede ser dividido en cuatro nodos hijos, y cada nodo hijo puede a su vez ser un nodo padre y ser dividido en otros cuatro nodos hijos. Un nodo hijo no dividido final, como un nodo hoja del árbol cuaternario, comprende un nodo de codificación, esto es, un bloque de vídeo codificado. Datos de sintaxis asociados con un flujo de bits codificado pueden definir un número máximo de veces que un bloque de árbol puede ser dividido, y también pueden definir un tamaño mínimo de los nodos de codificación.

Una CU incluye un nodo de codificación y unidades de predicción (PU) y unidades de transformación (TU) asociadas con el nodo de codificación. Un tamaño de la CU corresponde a un tamaño del nodo de codificación y debe ser de forma cuadrada. El tamaño de la CU puede oscilar de 8x8 píxeles hasta el tamaño del bloque de árbol con un máximo de 64x64 píxeles o mayor. Cada CU puede contener una o más PU y una o más TU. Datos de sintaxis asociados con una CU pueden describir, por ejemplo, la partición de las CU en una o más PU. Los modos de partición pueden diferir entre si la CU es codificada en modo salto o directo, codificada en modo de intrapredicción, o codificada en modo de interpredicción. Las PU pueden ser particionadas para que no sean de forma cuadrada. Los datos de sintaxis asociados con una CU también pueden describir, por ejemplo, la partición de la CU en una o más TU de acuerdo con un árbol cuaternario. Una TU puede ser de una forma cuadrada o no cuadrada.

El estándar HEVC permite transformaciones de acuerdo con las TU, que pueden ser diferentes para diferentes CU. Las TU son típicamente dimensionadas basándose en el tamaño de las PU dentro de una CU dada definida para una LCU particionada, aunque esto puede no ser siempre el caso. Las TU son típicamente del mismo tamaño o más pequeñas que las PU. En algunos ejemplos, muestras residuales que corresponden a una CU pueden ser subdivididas en unidades más pequeñas usando una estructura de árbol cuaternario conocida como “árbol cuaternario residual” (RQT). A los nodos hoja del RQT se puede hacer referencia como TU. Valores de diferencia de píxeles asociados con las TU pueden ser transformados para producir coeficientes de transformación, los cuales pueden ser cuantizados.

En general, una PU incluye datos relativos al proceso de predicción. Por ejemplo, cuando la PU es codificada por intramodo, la PU puede incluir datos que describen un modo de intrapredicción para la PU. Como otro ejemplo, cuando la PU es codificada por intermodo, la PU puede incluir datos que definen un vector 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 (p. ej., precisión de un cuarto de píxel o precisión de un octavo de píxel), una imagen de referencia a la cual el vector de movimiento apunta, y/o una lista de imagen de referencia (p. ej., Lista 0, Lista 1, o Lista C) para el vector de movimiento.

En general, una TU es usada para los procesos de transformación y cuantización. Una CU dada que tiene una o más PU también puede incluir una o más unidades de transformación (TU). Tras la predicción, el codificador de vídeo 20 puede calcular valores residuales que corresponden a la PU. Los valores residuales comprenden valores de diferencia

de píxeles que pueden ser transformados en coeficientes de transformación, cuantizados, y escaneados usando las TU para producir coeficientes de transformación serializados para la codificación por entropía. Esta divulgación usa típicamente el término “bloque de vídeo” para referirse a un nodo de codificación (esto es, bloque de codificación) de una CU. En algunos casos específicos, esta divulgación también puede usar el término “bloque de vídeo” para referirse a un bloque de árbol, esto es, LCU, o una CU, lo cual incluye un nodo de codificación y PU y TU.

Una secuencia de vídeo incluye típicamente una serie de fotogramas o imágenes de vídeo. Un grupo de imágenes (GOP) comprende generalmente una serie de una o más de las imágenes de vídeo. Un GOP puede incluir datos de sintaxis en una cabecera del GOP, una cabecera de una o más de las imágenes, u otro lugar, que describe un número de imágenes en el GOP. Cada segmento de una imagen incluye típicamente datos de sintaxis de segmento que describen un modo de codificación para el segmento respectivo. El codificador de vídeo 20 opera típicamente en bloques de vídeo dentro de segmentos 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 de acuerdo con un estándar de codificación especificado.

Como un ejemplo, el HM soporta predicción en varios tamaños de PU. Asumiendo que el tamaño de una CU particular es 2Nx2N, el HM soporta intrapredicción en tamaños de PU de 2Nx2N o NxN, e interpredicción en tamaños de PU simétricos de 2Nx2N, 2NxN, Nx2N, o NxN. El HM también soporta partición asimétrica para interpredicción en tamaños de PU de 2NxnU, 2NxnD, nLx2N, y nRx2N. En partición asimétrica, una dirección de una CU no está particionada, mientras que la otra dirección está particionada en 25 % y 75 %. La porción de la CU que corresponde a la partición de 25 % está indicada por una “n” seguida de una indicación de “Up”, “Down”, “Left”, o “Right”. Por lo tanto, por ejemplo, “2NxnU” se refiere a una CU 2Nx2N que está particionada horizontalmente con una PU 2Nx0.5N por arriba y una PU 2Nx1,5N por abajo.

En esta divulgación, “NxN” y “N por N” pueden ser usados intercambiablemente para referirse a las dimensiones de píxel de un bloque de vídeo en términos de las dimensiones vertical y horizontal, p. ej., 16 x 16 píxeles o 16 por 16 píxeles. En general, un bloque 16x16 tiene 16 píxeles en una dirección vertical (y = 16) y 16 píxeles en una dirección horizontal (x = 16). De igual modo, un bloque NxN tiene generalmente N píxeles en una dirección vertical y N píxeles en una dirección horizontal, donde N representa un valor entero no negativo. Los píxeles en un bloque pueden estar dispuestos en filas y columnas. Además, los bloques no necesitan necesariamente tener el mismo número de píxeles en la dirección horizontal como en la dirección vertical. Por ejemplo, los bloques pueden comprender NxM píxeles, donde M no es necesariamente igual a N.

Después de la codificación intrapredictiva o interpredictiva usando las PU de una CU, el codificador de vídeo 20 puede calcular datos residuales para las TU de la CU. Las PU pueden comprender datos de píxel en el dominio espacial (también referido como el dominio de píxel) y las TU pueden comprender coeficientes en el dominio de transformación después de la aplicación de una transformación, p. ej., una transformación de coseno discreta (DCT), una transformación de entero, una transformación de ondícula, o una transformación conceptualmente similar a los datos de vídeo residuales. Los datos residuales pueden corresponder a diferencias de píxel entre los píxeles de la imagen no codificada y los valores de predicción que corresponden a las PU. El codificador de vídeo 20 puede formar las TU que incluyen los datos residuales para la CU, y luego transformar las TU para producir coeficientes de transformación para la CU.

Después de cualquier transformación para producir coeficientes de transformación, el codificador de vídeo 20 puede realizar cuantización de los coeficientes de transformación. La cuantización se refiere generalmente a un proceso en el cual los coeficientes de transformación son cuantizados para reducir posiblemente la cantidad de datos usados para representar los coeficientes, proporcionando compresión adicional. El proceso de cuantización puede reducir la profundidad de bit asociada con algunos o todos los coeficientes. Por ejemplo, un valor n-bit puede ser redondeado hacia abajo hasta un valor m-bit durante la cuantización, donde n es mayor que m.

En algunos ejemplos, el codificador de vídeo 20 puede usar un orden de escaneo predefinido para escanear los coeficientes de transformación cuantizados para producir un vector serializado que puede ser codificado por entropía. En otros ejemplos, el codificador de vídeo 20 puede realizar un escaneo adaptativo. Tras escanear los coeficientes de transformación cuantizados para formar un vector unidimensional, el codificador de vídeo 20 puede codificar por entropía el vector unidimensional, p. ej., de acuerdo con codificación adaptativa según el contexto de longitud variable (CAVLC), codificación aritmética binaria adaptativa al contexto (CABAC), codificación aritmética binaria adaptativa al contexto basada en sintaxis (SBAC), 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 elementos de sintaxis asociados con los datos de vídeo codificados para el uso por parte del decodificador de vídeo 30 en la decodificación de los datos de vídeo.

Para realizar CABAC, el codificador de vídeo 20 puede asignar un contexto dentro de un modelo de contexto a un símbolo a ser transmitido. El contexto puede referirse a, por ejemplo, si los valores vecinos del símbolo son distintos de cero o no. Para realizar CAVLC, el codificador de vídeo 20 puede seleccionar un código de longitud variable para un símbolo a ser transmitido. Las palabras de código en la codificación de longitud variable (VLC) pueden ser construidas de tal modo que los códigos relativamente más cortos corresponden a símbolos más probables, mientras

que los códigos más largos corresponden a símbolos menos probables. De esta forma, el uso de VLC puede lograr ahorros de un bit al, por ejemplo, usar palabras de código de igual longitud para cada símbolo a ser transmitido. La determinación de probabilidad puede estar basada en un contexto asignado al símbolo.

El codificador de vídeo 20 puede transmitir un flujo de bits que incluye una secuencia de bits que forma una representación de imágenes codificadas y datos asociados. El término “flujo de bits” puede ser un término colectivo usado para referirse a o bien un flujo de unidades de capa de abstracción de red (NAL) (p. ej., una secuencia de unidades NAL) o un flujo de bytes (p. ej., una encapsulación de un flujo de unidades NAL que contienen prefijos de código de inicio y unidades NAL como está especificado por el Anexo B del estándar HEVC). Una unidad NAL es una estructura de sintaxis que contiene una indicación del tipo de datos en la unidad NAL y bytes que contienen esos datos en la forma de una carga útil de secuencia de bytes sin procesar (RBSP) entremezclados según sea necesario con bits de prevención de emulación. Cada una de las unidades NAL puede incluir una cabecera de unidad NAL y puede encapsular una RBSP. La cabecera de unidad NAL puede incluir un elemento de sintaxis que indica un código de tipo de unidad NAL. El código de tipo de unidad NAL especificado por la cabecera de unidad NAL de una unidad NAL indica el tipo de la unidad NAL. Una RBSP puede ser una estructura de sintaxis que contiene un número entero de bytes que está encapsulada dentro de una unidad NAL. En algunos casos, una RBSP incluye cero bits.

Diferentes tipos de unidades NAL pueden encapsular diferentes tipos de RBSP. Por ejemplo, un primer tipo de unidad NAL puede encapsular una RBSP para un PPS, un segundo tipo de unidad NAL puede encapsular una RBSP para un segmento de segmento, un tercer tipo de unidad NAL puede encapsular una RBSP para SEI, y así sucesivamente. A las unidades NAL que encapsulan RBSP para datos de codificación de vídeo (a diferencia de las RBSP para conjuntos de parámetros y mensajes SEI) se puede hacer referencia como unidades NAL de capa de codificación de vídeo (VCL). A las unidades NAL que contienen conjuntos de parámetros (p. ej., VPS, SPS, PPS, etc.) se puede hacer referencia como unidades NAL de conjuntos de parámetros.

Esta divulgación puede referirse a una unidad NAL que encapsula una RBSP para un segmento de segmento como una unidad NAL de segmento codificado. Como está definido en el HEVC WD, un segmento de segmento es un número entero de CTU ordenadas consecutivamente en escaneo de mosaico y contenidas en una única unidad NAL. En contraste, en el HEVC WD, un segmento puede ser un número entero de CTU contenidas en un segmento de segmento independiente y todos los segmentos de segmento subsiguientes (si hay alguno) que preceden al siguiente segmento de segmento independiente (si hay alguno) dentro de la misma unidad de acceso. Un segmento de segmento independiente es un segmento de segmento para el cual los valores de los elementos de sintaxis de la cabecera de segmento de segmento no son inferidos de los valores para un segmento de segmento precedente. Un segmento de segmento dependiente es un segmento de segmento para el cual los valores de algunos elementos de sintaxis de la cabecera de segmento de segmento son inferidos de los valores para el segmento de segmento independiente precedente en orden de decodificación. La RBSP de una unidad NAL de segmento codificado puede incluir una cabecera de segmento de segmento y datos de segmento. Una cabecera de segmento de segmento es una parte de un segmento de segmento codificado que contiene los elementos de datos que pertenecen a la primera o todas las CTU representadas en el segmento de segmento. Una cabecera de segmento es una cabecera de segmento de segmento del segmento de segmento independiente que es un segmento de segmento actual o el segmento de segmento independiente más reciente que precede a un segmento de segmento dependiente actual en orden de decodificación.

Una VPS es una estructura de sintaxis que comprende elementos de sintaxis que se aplican a cero o más secuencias de vídeo codificado (CVS) enteras. Una SPS es una estructura de sintaxis que contiene elementos de sintaxis que se aplican a cero o más CVS enteras. Una SPS puede incluir un elemento de sintaxis que identifica una VPS que está activa cuando la SPS está activa. Por lo tanto, los elementos de sintaxis de una VPS pueden ser más generalmente aplicables que los elementos de sintaxis de una SPS.

Un conjunto de parámetros (p. ej., una VPS, SPS, PPS, etc.) puede contener una identificación que es referenciada, directamente o indirectamente, a partir de una cabecera de segmento de un segmento. El proceso de referenciado es conocido como “activación”. Por lo tanto, cuando el decodificador de vídeo 30 está decodificando un segmento particular, se dice que un conjunto de parámetros referenciado, directamente o indirectamente, por un elemento de sintaxis en una cabecera de segmento del segmento particular está “activado”. Dependiendo del tipo de conjunto de parámetros, la activación puede ocurrir por imagen o por secuencia. Por ejemplo, una cabecera de segmento de un segmento puede incluir un elemento de sintaxis que identifica una PPS. Por lo tanto, cuando un codificador de vídeo codifica el segmento, la PPS puede estar activada. Además, la PPS puede incluir un elemento de sintaxis que identifica una SPS. Por lo tanto, cuando una PPS que identifica la SPS está activada, la SPS puede estar activada. La SPS puede incluir un elemento de sintaxis que identifica una VPS. Por lo tanto, cuando una SPS que identifica la VPS está activada, la VPS está activada.

El decodificador de vídeo 30 puede recibir un flujo de bits generado por el codificador de vídeo 20. Además, el decodificador de vídeo 30 puede analizar el flujo de bits para obtener elementos de sintaxis del flujo de bits. El decodificador de vídeo 30 puede reconstruir las imágenes de los datos de vídeo basándose al menos en parte en los elementos de sintaxis obtenidos del flujo de bits. El proceso para reconstruir los datos de vídeo puede ser generalmente recíproco al proceso realizado por el codificador de vídeo 20. Por ejemplo, el decodificador de vídeo 30

puede usar vectores de movimiento de las PU para determinar bloques predictivos para las PU de una CU actual. Además, el decodificador de vídeo 30 puede cuantizar inversamente bloques de coeficientes de las TU de la CU actual. El decodificador de vídeo 30 puede realizar transformaciones inversas en los bloques de coeficientes para reconstruir bloques de transformación de las TU de la CU actual. El decodificador de vídeo 30 puede reconstruir los bloques de codificación de la CU actual añadiendo las muestras de los bloques predictivos para las PU de la CU actual a muestras correspondientes de los bloques de transformación de las TU de la CU actual. Reconstruyendo los bloques de codificación para cada CU de una imagen, el decodificador de vídeo 30 puede reconstruir la imagen.

En el HEVC WD, una CVS puede iniciarse a partir de una imagen de refresco de decodificación instantánea (IDR), o una imagen de acceso de enlace roto (BLA), o una imagen de acceso aleatorio limpio (CRA) que es la primera imagen en el flujo de bits, incluidas todas las subsiguientes imágenes que no son imágenes IDR o BLA. Una imagen IDR contiene solo segmentos I (esto es, segmentos en los cuales solo es usada la intrapredicción). Una imagen IDR puede ser la primera imagen en el flujo de bits en orden de decodificación, o puede aparecer después en el flujo de bits. Cada imagen IDR es la primera imagen de una CVS en orden de decodificación. En el HEVC WD, una imagen IDR puede ser una imagen de punto de acceso intraaleatorio (IRAP) para la cual cada unidad NAL VCL tiene un nal_unit_type igual a IDR_WRADL o IDR_N_LP.

Imágenes IDR pueden ser usadas para el acceso aleatorio. Sin embargo, las imágenes que siguen a una imagen IDR en orden de decodificación no pueden usar imágenes decodificadas antes de la imagen IDR como referencia. Correspondientemente, los flujos de bits que se basan en imágenes IDR para el acceso aleatorio pueden tener una eficiencia de codificación significativamente más baja que los flujos de bits que usan tipos adicionales de imágenes de acceso aleatorio. En al menos algunos ejemplos, una unidad de acceso IDR es una unidad de acceso que contiene una imagen IDR.

El concepto de imágenes CRA fue introducido en HEVC para permitir que las imágenes que siguen a una imagen CRA en orden de decodificación, pero preceden a la imagen CRA en orden de salida, usen imágenes decodificadas antes de la imagen CRA para la referencia. A las imágenes que siguen a una imagen CRA en orden de decodificación, pero preceden a la imagen CRA en orden de salida, se hace referencia como imágenes iniciales asociadas con la imagen CRA (o imágenes iniciales de la imagen CRA). Esto es, para mejorar la eficiencia de codificación, el concepto de imágenes CRA fue introducido en HEVC para permitir que las imágenes que siguen a una imagen CRA en orden de decodificación pero preceden a la imagen CRA en orden de salida usen imágenes decodificadas antes de la imagen CRA para la referencia. Una unidad de acceso CRA es una unidad de acceso en la cual la imagen codificada es una imagen CRA. En el HEVC WD, una imagen CRA es una imagen de acceso intraaleatorio para la cual cada unidad NAL VCL tiene un nal_unit_type igual a CRA_NUT.

Las imágenes iniciales de una imagen CRA son correctamente decodificables si la decodificación comienza desde una imagen IDR o una imagen CRA que se da antes de la imagen CRA en orden de decodificación. Sin embargo, las imágenes iniciales de una imagen CRA pueden ser no decodificables cuando se da acceso aleatorio desde la imagen CRA. Por consiguiente, un decodificador de vídeo decodifica típicamente las imágenes iniciales de una imagen CRA durante la decodificación de acceso aleatorio. Para prevenir la propagación de errores desde las imágenes de referencia que pueden no estar disponibles dependiendo de si la decodificación comienza, ninguna imagen que sigue a una imagen CRA tanto en orden de decodificación como orden de salida puede usar ninguna imagen que preceda a la imagen CRA ni en orden de decodificación ni orden de salida (lo cual incluye a las imágenes iniciales) para la referencia.

El concepto de una imagen BLA fue introducido en HEVC tras la introducción de imágenes CRA y está basado en el concepto de imágenes CRA. Una imagen BLA se origina típicamente a partir de un empalme de flujo de bits en la posición de una imagen CRA, y en el flujo de bits empalmado, la imagen CRA de punto de empalme es cambiada a una imagen BLA. Por lo tanto, las imágenes BLA pueden ser imágenes CRA en los flujos de bits originales y una imagen CRA es cambiada para ser una imagen BLA por el empalmador de flujo de bits tras el empalme de flujo de bits en la ubicación de la imagen CRA. En algunos ejemplos, a una unidad de acceso que contiene una imagen RAP se puede hacer referencia aquí como una unidad de acceso RAP. Una unidad de acceso BLA es una unidad de acceso que contiene una imagen BLA. En el HEVC WD, una imagen BLA puede ser una imagen de acceso intraaleatorio para la cual cada unidad NAL VCL tiene nal_unit_type igual a BLA_W_LP, BLA_W_RADL, o BLA_N_LP.

En general, una imagen IRAP contiene solo segmentos I, y puede ser una imagen BLA, una imagen CRA, o una imagen IDR. Por ejemplo, el HEVC WD indica que una imagen IRAP puede ser una imagen codificada para la cual cada unidad NAL VCL tiene nal_unit_type en el rango de BLA_W_LP a RSV_IRAP_VCL23, inclusive. Además, el HEVC WD indica que la primera imagen en el flujo de bits en orden de decodificación debe ser una imagen IRAP. La tabla 7-1 de HEVC WD muestra los códigos de tipo de unidad NAL y las clases de tipo de unidad NAL. La tabla 7-1 de HEVC WD está reproducida abajo.

Tabla 7-1 - Códigos de tipo de unidad NAL y clases de tipo de unidad NAL

nal_unit_ type	Nombre de nal_unit_type	Contenido de unidad NAL y estructura de sintaxis de RBSP	Clase de tipo de unidad NAL
0 1	TRAIL N TRAIL R	Segmento de segmento codificado de una slice_segment_layer_rbsp() de imagen posterior no TSA, no STSA	VCL
2 3	TSA_N TSA_R	Segmento de segmento codificado de una slice_segment_layer_rbsp() de imagen TSA	VCL
4 5	STSA_N STSA_R	Segmento de segmento de una slice_segment_layer_rbsp() de imagen STSA	VCL
6 7	RADL_N RADL_R	Segmento de segmento codificado de una slice_segment_layer_rbsp() de imagen RADL	VCL
8 9	RASL_N RASL_R	Segmento de segmento codificado de una slice_segment_layer_rbsp() de imagen RASL	VCL
10 12 14	RSV_VCL_N10 RSV_VCL_N12 RSV_VCL_N14	Tipos de unidad NAL VCL de no referencia de subcapa no IRAP reservados	VCL
11 13 15	RSV_VCL_R11 RSV_VCL_R13 RSV_VCL_R15	Tipos de unidad NAL VCL de referencia de subcapa no IRAP reservados	VCL
16 17 18	BLA_W_LP BLA_W_RADL BLA_N_LP	Segmento de segmento codificado de una slice_segment_layer_rbsp() de imagen BLA	VCL
19 20	IDR_W_RADL IDR_N_LP	Segmento de segmento codificado de una imagen IDR	VCL
		slice_segment_layer_rbsp()	
21	CRA_NUT	Segmento de segmento codificado de una slice_segment_layer_rbsp() de imagen CRA	VCL
22 23	RSV_IRAP_VCL22 RSV_IRAP_VCL23	Tipos de unidad NAL VCL de IRAP reservados	VCL
24..31	RSV_VCL24.. RSV_VCL31	Tipos de unidad NAL VCL de no IRAP reservados	VCL
32	VPS_NUT	video_parameter_set_rbsp() de conjunto de parámetros de vídeo	no VCL
33	SPS_NUT	seq_parameter_set_rbsp() de conjunto de parámetros de secuencia	no VCL
34	PPS_NUT	pic_parameter_set_rbsp() de conjunto de parámetros de imagen	no VCL
35	AUD_NUT	access_unit_delimiter_rbsp() de delimitador de unidad de acceso	no VCL
36	EOS_NUT	end_of_seq_rbsp() de fin de secuencia	no VCL
37	EOB_NUT	end_of_bitstream_rbsp() de fin de flujo de bits	no VCL
38	FD_NUT	filler_data_rbsp() de datos de relleno	no VCL
39 40	PREFIX_SEI_NUT SUFFIX_SEI_NUT	sei_rbsp() de información de mejora suplementaria	no VCL
41..47	RSV_NVCL41.. RSV_NVCL47	Reservado	no VCL
48..63	UNSPEC48.. UNSPEC63	Sin especificar	no VCL

Una diferencia entre las imágenes BLA y las imágenes CRA es como sigue. Para una imagen CRA, las imágenes iniciales asociadas son correctamente decodificables si la decodificación comienza desde una imagen RAP antes de la imagen CRA en orden de decodificación. Sin embargo, las imágenes iniciales asociadas con una imagen CRA pueden no ser correctamente decodificables cuando se da acceso aleatorio desde la imagen CRA (esto es, cuando la decodificación comienza desde la imagen CRA, o en otras palabras, cuando la imagen CRA es la primera imagen en el flujo de bits). En contraste, puede no haber ningún escenario donde las imágenes iniciales asociadas con una

imagen BLA sean decodificables, incluso cuando la decodificación comienza desde una imagen RAP antes de la imagen BLA en orden de decodificación.

Algunas de las imágenes iniciales asociadas con una imagen CRA particular o una imagen BLA particular pueden ser correctamente decodificables incluso cuando la imagen CRA particular o la imagen BLA particular es la primera imagen en un flujo de bits. A estas imágenes iniciales se puede hacer referencia como imágenes iniciales codificables (DLP) o imágenes iniciales decodificables de acceso aleatorio (RADL). En el HEVC WD, una imagen RADL puede ser una imagen codificada para la cual cada unidad NAL VCL tiene un nal_unit_type igual a RADL_R o RADL_N. Además, el HEVC WD indica que todas las imágenes RADL son imágenes iniciales y que las imágenes RADL no son usadas como imágenes de referencia para el proceso de decodificación de imágenes posteriores de la misma imagen IRAP asociada. Cuando están presentes, todas las imágenes RADL preceden, en orden de decodificación, a todas las imágenes posteriores de la misma imagen IRAP asociada. El HEVC WD indica que una unidad de acceso RADL puede ser una unidad de acceso en la cual la imagen codificada es una imagen RADL. Una imagen posterior puede ser una imagen que sigue a la imagen IRAP asociada (esto es, la imagen IRAP previa en orden de decodificación) en orden de salida.

A otras imágenes iniciales se puede hacer referencia como imágenes iniciales no decodificables (NLP) o imágenes iniciales omitidas de acceso aleatorio (RASL). En el HEVC WD, una imagen RASL puede ser una imagen codificada para la cual cada unidad NAL VCL tiene un nal_unit_type igual a RASL_R o RASL_N. Todas las imágenes RASL son imágenes iniciales de una imagen BLA o CRA asociada.

Siempre que los conjuntos de parámetros necesarios estén disponibles cuando ellos necesitan ser activados, una imagen IRAP y todas las imágenes no RASL subsiguientes en orden de decodificación pueden ser correctamente decodificadas sin realizar el proceso de decodificación de ningunas imágenes que preceden a la imagen IRAP en orden de decodificación. Puede haber imágenes en un flujo de bits que contienen solo segmentos I que no son imágenes IRAP.

En la codificación de vídeo multivista, puede haber múltiples vistas de la misma escena desde diferentes puntos de vista. El término “unidad de acceso” puede ser usado para referirse al conjunto de imágenes que corresponden a la misma instancia de tiempo. Por lo tanto, los datos de vídeo pueden ser conceptualizados como una serie de unidades de acceso que se dan a lo largo del tiempo. Un “componente de vista” puede ser una representación codificada de una vista en una única unidad de acceso. En esta divulgación, una “vista” puede referirse a una secuencia o conjunto de componentes de vista asociados con el mismo identificador de vista. Un componente de vista puede contener un componente de vista de textura y un componente de vista de profundidad. En esta divulgación, una “vista” puede referirse a un conjunto o secuencia de uno o más componentes de vista asociados con el mismo identificador de vista.

Un componente de vista de textura (esto es, una imagen de textura) puede ser una representación codificada de la textura de una vista en una única unidad de acceso. Una vista de textura puede ser una secuencia de componentes de vista de textura asociados con un valor idéntico de un índice de orden de vista. Un índice de orden de vista de una vista puede indicar una posición de cámara de la vista en relación con otras vistas. Un componente de vista de profundidad (esto es, una imagen de profundidad) puede ser una representación codificada de la profundidad de una vista en una única unidad de acceso. Una vista de profundidad puede ser un conjunto o secuencia de uno o más componentes de vista de profundidad asociados con un valor idéntico de índice de orden de vista.

En MV-HEVC, 3D-HEVC y SHVC, un codificador de vídeo puede generar un flujo de bits que comprende una serie de unidades NAL. Diferentes unidades NAL del flujo de bits pueden estar asociadas con diferentes capas del flujo de bits. Una capa puede ser definida como un conjunto de unidades NAL VCL y unidades NAL no VCL asociadas que tienen el mismo identificador de capa. Una capa puede ser equivalente a una vista en la codificación de vídeo multivista. En la codificación de vídeo multivista, una capa puede contener todos los componentes de vista de la misma capa con diferentes instancias de tiempo. Cada componente de vista puede ser una imagen codificada de la escena de vídeo que pertenece a una vista específica en una instancia de tiempo específica. En algunos ejemplos de la codificación de vídeo 3D, una capa puede contener o bien todas las imágenes de profundidad codificadas de una vista específica o imágenes de textura codificadas de una vista específica. En otros ejemplos de la codificación de vídeo 3D, una capa puede contener tanto componentes de vista de textura como componentes de vista de profundidad de una vista específica. Similarmente, en el contexto de la codificación de vídeo escalable, una capa corresponde típicamente a imágenes codificadas que tienen características de vídeo diferentes de las imágenes codificadas en otras capas. Tales características de vídeo incluyen típicamente resolución espacial y nivel de calidad (p. ej., relación señal-ruido). En HEVC y sus extensiones, la escalabilidad temporal puede ser alcanzada dentro de una capa definiendo un grupo de imágenes con un nivel temporal particular como una subcapa.

Para cada capa respectiva del flujo de bits, los datos en una capa inferior pueden ser decodificados sin referencia a los datos en cualquier capa superior. En la codificación de vídeo escalable, por ejemplo, los datos en una capa base pueden ser decodificados sin referencia a los datos en una capa de mejora. En general, las unidades NAL solo pueden encapsular datos de una única capa. Por lo tanto, las unidades NAL que encapsulen datos de la capa restante más alta del flujo de bits pueden ser eliminadas del flujo de bits sin afectar la decodificabilidad de los datos en las capas restantes del flujo de bits. En la codificación multivista y 3D-HEVC, las capas más altas pueden incluir componentes

de vista adicionales. En SHVC, las capas más altas pueden incluir datos de mejora de relación señal-ruido (SNR), datos de mejora espacial, y/o datos de mejora temporal. En MV-HEVC, 3D-HEVC y SHVC, a una capa se puede hacer referencia como una “capa base” si un decodificador de vídeo puede decodificar imágenes en la capa sin referencia a los datos de ninguna otra capa. La capa base puede adecuarse a la especificación base HEVC (p. ej., HEVC WD).

En SVC, a las capas distintas de la capa base se puede hacer referencia como “capas de mejora” y pueden proporcionar información que mejora la calidad visual de los datos de vídeo decodificados del flujo de bits. SVC puede mejorar la resolución espacial, la relación señal-ruido (esto es, calidad) o la tasa temporal. En la codificación de vídeo escalable (p. ej., SHVC), una “representación de capa” puede ser una representación codificada de una capa espacial en una única unidad de acceso. Para facilidad de explicación, esta divulgación puede referirse a los componentes de vista y/o representaciones de capa como “componentes de vista/representaciones de capa” o simplemente “imágenes”.

Para implementar las capas en HEVC, las cabeceras de las unidades NAL incluyen un elemento de sintaxis nuh_layer_id, al cual se hizo previamente referencia como el elemento de sintaxis nuh_reserved_zero_6bits en varios borradores de trabajo que precedieron al estándar HEVC final. En el estándar HEVC base, el elemento de sintaxis nuh layer id está limitado a un valor de 0. Sin embargo, en MV-HEVC, 3D-HEVC, y SVC, el elemento de sintaxis nuh layer id puede ser mayor que 0 para especificar un identificador de una capa. Las unidades NAL de un flujo de bits que tienen elementos de sintaxis nuh_layer_id que especifican diferentes valores pertenecen a diferentes capas del flujo de bits.

En algunos ejemplos, el elemento de sintaxis nuh_layer_id de una unidad NAL es igual a 0 si la unidad NAL se refiere a una capa base en la codificación multivista (p. ej., MV-HEVC), la codificación 3DV (p. ej., 3D-HEVC), o la codificación de vídeo escalable (p. ej., SHVC). Los datos en una capa base de un flujo de bits pueden ser decodificados sin referencia a los datos en ninguna otra capa del flujo de bits. Si una unidad NAL no se refiere a una capa base en la codificación multivista, 3DV, o la codificación de vídeo escalable, el elemento de sintaxis nuh layer id de la unidad NAL puede tener un valor distinto de cero.

Además, algunos componentes de vista/representaciones de capa dentro de una capa pueden ser decodificados sin referencia a otros componentes de vista/representaciones de capa dentro de la misma capa. Por lo tanto, las unidades NAL que encapsulan los datos de ciertos componentes de vista/representaciones de capa de una capa pueden ser eliminados del flujo de bits sin afectar la decodificabilidad de otros componentes de vista/representaciones de capa en la capa. Eliminar unidades NAL que encapsulan datos de tales componentes de vista/representaciones de capa puede reducir la tasa de fotogramas del flujo de bits. A un subconjunto de componentes de vista/representaciones de capa dentro de una capa que pueden ser decodificados sin referencia a otros componentes de vista/representaciones de capa dentro de la capa se puede hacer referencia aquí como una “subcapa” o una “subcapa temporal”.

Las unidades NAL pueden incluir elementos de sintaxis temporal_id que especifican identificadores temporales (esto es, TemporalIds) de las unidades NAL. El identificador temporal de una unidad NAL identifica una subcapa a la cual pertenece la unidad NAL. Por lo tanto, cada subcapa de un flujo de bits puede tener un identificador temporal diferente. En general, si el identificador temporal de una primera unidad NAL de una capa es menor que el identificador temporal de una segunda unidad NAL de la misma capa, los datos encapsulados por la primera unidad NAL pueden ser decodificados sin referencia a los datos encapsulados por la segunda unidad NAL.

Un flujo de bits puede estar asociado con una pluralidad de puntos de operación. Cada punto de operación de un flujo de bits está asociado con un conjunto de identificadores de capa (p. ej., un conjunto de valores nuh_layer_id) y un identificador temporal. El conjunto de identificadores de capa puede ser designado como OpLayerIdSet y el identificador temporal puede ser designado como TemporalID. Si el identificador de capa de una unidad NAL está en un conjunto de identificadores de capa de punto de operación y el identificador temporal de la unidad NAL es menor que o igual al identificador temporal de punto de operación, la unidad NAL está asociada con el punto de operación. Por lo tanto, un punto de operación puede corresponder a un subconjunto de unidades NAL en el flujo de bits. HEVC define un punto de operación como un flujo de bits creado a partir de otro flujo de bits mediante la operación del proceso de extracción de subflujo de bits con el otro flujo de bits, un TemporalID máximo objetivo, y una lista de identificador de capa objetivo como insumos.

Como está presentado arriba, esta divulgación se refiere al almacenamiento de contenido de vídeo en un archivo basado en el formato de archivo multimedia base ISO (ISOBMFF). En particular, esta divulgación describe varias técnicas para almacenar flujos de vídeo que contienen múltiples capas codificadas, en donde cada capa puede ser una capa escalable, una vista de textura, una vista de profundidad, u otros tipos de capas o vistas. Las técnicas de esta divulgación pueden ser aplicadas a, por ejemplo, almacenamiento de datos de vídeo MV-HEVC, datos de vídeo SHVC, datos de vídeo 3D-HEVC, y/u otros tipos de datos de vídeo.

Formatos de archivo y estándares de formato de archivo serán ahora brevemente comentados. Los estándares de formato de archivo incluyen formato de archivo multimedia base ISO (ISOBMFF, ISO/IEC 14496-12, en adelante, “ISO/IEC 14996-12”) y otros estándares de formato de archivo derivados del ISOBMFF, incluidos el formato de archivo MPEG-4 (ISO/IEC 14496-14), el formato de archivo 3GPP (3GPP TS 26.244) y el formato de archivo AVC (ISO/IEC

14496-15, en adelante “ISO/IEC 14996-15”). Por lo tanto, ISO/IEC 14496-12 especifica el formato de archivo multimedia base ISO. Otros documentos amplían el formato de archivo multimedia base ISO para aplicaciones específicas. Por ejemplo, ISO/IEC 14496-15 describe el transporte de vídeo estructurado de unidad NAL en el formato de archivo multimedia base ISO. H.264/AVC y HEVC, así como sus extensiones, son ejemplos de vídeo estructurado de unidad NAL. ISO/IEC 14496-15 incluye secciones que describen el transporte de unidades NAL H.264/AVC. Adicionalmente, la sección 8 de ISO/IEC 14496-15 describe el transporte de unidades NAL HEVC.

El ISOBMFF es usado como la base para muchos formatos de encapsulación de códec, tal como el formato de archivo AVC, así como para muchos formatos contenedores multimedia, tal como el formato de archivo MPEG-4, el formato de archivo 3GPP (3GP), y el formato de archivo DVB. Además de medios continuos, tales como audio y vídeo, medios estáticos, tales como imágenes, así como metadatos, pueden ser almacenados en un archivo conforme a ISOBMFF. Los archivos estructurados de acuerdo con ISOBMFF pueden ser usados para muchos fines, incluidos la carga útil de archivo multimedia local, la descarga progresiva de un archivo remoto, segmentos para la transmisión adaptativa dinámica por HTTP (DASH), contenedores para contenido a ser transmitido y sus instrucciones de paquetización, y la grabación de transmisiones multimedia en tiempo real recibidas. Por lo tanto, aunque originalmente está diseñado para el almacenamiento, el ISOBMFF ha demostrado ser valioso para la transmisión, p. ej. para la descarga progresiva o DASH. Para fines de transmisión, pueden ser usados los segmentos de película definidos en ISOBMFF.

Un archivo que se adecua al formato de archivo HEVC puede comprender una serie de objetos, llamados cajas. Una caja puede ser un bloque de construcción orientado al objeto definido por un identificador de tipo único y una longitud. Por ejemplo, una caja puede ser la estructura de sintaxis elemental en el ISOBMFF, incluido un tipo de caja codificada de cuatro caracteres, un recuento de bytes de la caja, y una carga útil. En otras palabras, una caja puede ser una estructura de sintaxis que comprende un tipo de caja codificada, un recuento de bytes de la caja, y una carga útil. En algunos ejemplos, todos los datos en un archivo que se adecua al formato de archivo HEVC pueden ser contenidos dentro de cajas y puede no haber ningún dato en el archivo que no está en una caja. Por lo tanto, un archivo ISOBMFF puede constar de una secuencia de cajas, y las cajas pueden contener otras cajas. Por ejemplo, la carga útil de una caja puede incluir una o más cajas adicionales. Las FIG. 5A, FIG. 5B, y FIG. 6, descritas en detalle en otro lugar en esta divulgación, muestran cajas de ejemplo dentro de un archivo, de conformidad con una o más técnicas de esta divulgación.

Un archivo que se adecua al ISOBMFF puede incluir varios tipos de cajas. Por ejemplo, un archivo que se adecua al ISOBMFF puede incluir una caja de tipo de archivo, una caja de datos de medios, una caja de película, una caja de fragmento de película, y así sucesivamente. En este ejemplo, una caja de tipo de archivo incluye un tipo de archivo e información de compatibilidad. Una caja de datos de medios puede contener muestras (p. ej., imágenes codificadas). Una caja de película (“moov”) contiene metadatos para transmisiones de medios continuos presentes en el archivo. Cada una de las transmisiones de medios continuos puede estar representada en el archivo como una pista. Por ejemplo, una caja de película puede contener metadatos sobre una película (p. ej., relaciones lógicas y de tiempo entre muestras, y también punteros a ubicaciones de muestras). Las cajas de películas pueden incluir varios tipos de subcajas. Las subcajas en una caja de película pueden incluir una o más cajas de pista. Una caja de pista puede incluir información sobre una pista individual de una película. Una caja de pista puede incluir una caja de cabecera de pista que especifica información general de una única pista. Además, una caja de pista puede incluir una caja de medios que contiene una caja de información de medios. La caja de información de medios puede incluir una caja de tabla de muestras que contiene indexación de datos de muestras de medios en la pista. La información en la caja de tabla de muestras puede ser usada para ubicar muestras en el tiempo y, para cada una de las muestras de la pista, un tipo, tamaño, contenedor, y compensación dentro de ese contenedor de la muestra. Por lo tanto, los metadatos para una pista están encerrados en una caja de pista (“track”), mientras que el contenido de medios de una pista está o bien encerrado en una caja de datos de medios (“mdat”) o directamente en un archivo separado. El contenido de medios para pistas comprende (p. ej., consta de) una secuencia de muestras, tales como unidades de acceso de audio o vídeo.

El ISOBMFF especifica los siguientes tipos de pistas: una pista de medios, la cual contiene un flujo de medios elemental, una pista de sugerencia, la cual incluye o bien instrucciones de transmisión de medios o representa un flujo de paquetes recibido, y una pista de metadatos temporizados, la cual comprende metadatos sincronizados en el tiempo. Los metadatos para cada pista incluyen una lista de entradas de descripción de muestra, proporcionando cada una el formato de codificación o encapsulación usado en la pista y los datos de inicialización necesarios para procesar ese formato. Cada muestra está asociada con una de las entradas de descripción de muestra de la pista.

El ISOBMFF habilita la especificación de metadatos específicos de muestra con varios mecanismos. Cajas específicas dentro de la caja de tabla de muestras (“stbl”) han sido estandarizadas para responder a las necesidades comunes. Por ejemplo, una caja de muestra de sincronización (“stss”) es una caja dentro de una caja de tabla de muestras. La caja de muestra de sincronización es usada para enlistar las muestras de acceso aleatorio de la pista. Esta divulgación se puede referir a una muestra enlistada por la caja de muestra de sincronización como una muestra de sincronización. En otro ejemplo, un mecanismo de agrupación de muestra habilita la asignación de muestras de acuerdo con un tipo de agrupación de cuatro caracteres en grupos de muestras que comparten la misma propiedad especificada como una entrada de descripción de grupo de muestra en el archivo. Varios tipos de agrupación han sido especificados en el ISOBMFF.

Una caja de tabla de muestras puede incluir una o más cajas SampleToGroup y una o más cajas de descripción de grupo de muestra (esto es, cajas SampleGroupDescription). Una caja SampleToGroup puede ser usada para determinar un grupo de muestra al cual pertenece una muestra, junto con una descripción asociada del grupo de muestra. En otras palabras, una caja SampleToGroup puede indicar un grupo al cual pertenece una muestra. Una caja SampleToGroup puede tener un tipo de caja de “sbgp”. Una caja SampleToGroup puede incluir un elemento de tipo de agrupación (p. ej., grouping_type). El elemento de tipo de agrupación puede ser un entero que identifica un tipo (esto es, un criterio usado para formar los grupos de muestra) de una agrupación de muestra. Además, una caja SampleToGroup puede incluir una o más entradas. Cada entrada en una caja SampleToGroup puede estar asociada con una serie no superpuesta diferente de muestras consecutivas en la pista. Cada entrada puede indicar un elemento de recuento de muestra (p. ej., sample_count) y un elemento de índice de descripción de grupo (p. ej., group_description_index). El elemento de recuento de muestra de una entrada puede indicar un número de muestras asociadas con la entrada. En otras palabras, el elemento de recuento de muestra de la entrada puede ser un entero que da el número de muestras consecutivas con el mismo descriptor de grupo de muestra. El elemento de índice de descripción de grupo puede identificar una caja SampleGroupDescription que contiene una descripción de las muestras asociadas con la entrada. Los elementos de índice de descripción de grupo de múltiples entradas pueden identificar la misma caja SampleGroupDescription.

Los diseños de formato de archivo actuales pueden tener uno o más problemas. Para almacenar contenido de vídeo de un códec de vídeo particular basado en el ISOBMFF, una especificación de formato de archivo para ese códec de vídeo puede ser necesaria. Para el almacenamiento de flujos de vídeo que contienen múltiples capas tales como MV-HEVC y SHVC, es posible reutilizar algunos de los conceptos del formato de archivo SVC y MVC. Sin embargo, muchas partes no pueden ser usadas directamente para flujos de vídeo SHVC y MV-HEVC. La aplicación directa del formato de archivo HEVC tiene al menos las siguientes deficiencias: los flujos de bits SHVC y MV-HEVC pueden comenzar con una unidad de acceso que contiene una imagen IRAP en la capa base, pero también pueden contener otras imágenes no IRAP en otras capas o viceversa. La muestra de sincronización no permite actualmente la indicación de un punto tal para el acceso aleatorio.

Esta divulgación describe soluciones potenciales a los problemas de arriba, así como proporciona otras mejoras potenciales, para habilitar el almacenamiento eficiente y flexible de flujos de vídeo que contienen múltiples capas. Las técnicas descritas en esta divulgación se aplican potencialmente a cualquier formato de archivo para el almacenamiento de tal contenido de vídeo codificado por cualquier códec de vídeo, aunque la descripción es específica para el almacenamiento de flujos de vídeo SHVC y MV-HEVC basados en el formato de archivo HEVC, el cual está especificado en la cláusula 8 de ISO/IEC 14496-15.

Una implementación de ejemplo de algunas técnicas de esta divulgación está descrita abajo. La implementación de ejemplo descrita abajo está basada en la última especificación integrada de 14496-15 en el documento de salida MPEG W13478. Los cambios al Anexo A (mostrados con subrayado) y las secciones añadidas (sección 9 para SHVC y sección 10 para MV-HEVC) están incluidos abajo. En otras palabras, ejemplos particulares de esta divulgación pueden modificar el Anexo A de ISO/IEC 14496-15 y pueden añadir las secciones 9 y/o 10 a ISO/IEC 14496-15. El texto mostrado con subrayado y doble subrayado puede ser de particular relevancia para los ejemplos de esta divulgación. Aunque el término SHVC está usado en varios lugares en los ejemplos descritos aquí, el diseño de esta divulgación no es en realidad solo para soportar el códec SHVC, sino que en cambio pueden ser soportados todos los códec multicapa, incluidos MV-HEVC, 3D-HEVC, a menos que se mencione explícitamente lo contrario.

La especificación ISOBMFF especifica seis tipos de puntos de acceso de flujo (SAP) para el uso con DASH. Los dos primeros tipos de SAP (tipos 1 y 2) corresponden a imágenes IDR en H.264/AVC y HEVC. El tercer tipo de SAP (tipo 3) corresponde a puntos de acceso aleatorios de GOP abierto por consiguiente imágenes BLA o CRA en HEVC. El cuarto tipo de SAP (tipo 4) corresponde a puntos de acceso aleatorios GDR.

En el formato de archivo L-HEVC actual, alguna información de alto nivel (p. ej., información de capas en el flujo de bits, tasa de bit, tasa de fotograma, subcapas temporales, paralelismo, puntos de operación, etc.) están señalados en LHEVCSampleEntry, HEVCLHVCSampleEntry, LHVCDecoderConfigurationRecord, info de contenido de pista (‘tcon’) y OperationPointsInformationBox (‘oinf’). En un ejemplo, el diseño de sintaxis de las cajas mencionadas arriba es como sigue:

Basándose en la estructura actual de las cajas de arriba, y la información contenida dentro de ellas, para reproducir el contenido en el archivo, un reproductor puede estar configurado para primero encontrar la caja ‘oinf (solo una en el archivo) para saber qué puntos de operación están incluidos, y luego elegir uno de los puntos de operación a ser reproducido. El reproductor de vídeo puede luego comprobar las cajas ‘tcon’ (una en cada pista que contiene vídeo L-HEVC) para saber qué pistas contienen las capas de los puntos de operación elegidos.

```
//LHVC and HEVCLHVC sample entry
class LHEVCConfigurationBox extends Box('lhvC') {
    LHEVCDecoderConfigurationRecord() LHEVCConfig;
}

class HEVCLHVCSampleEntry() extends HEVCSampleEntry() {
    LHEVCConfigurationBox          lhvcconfig;
    MPEG4BitRateBox ();                        // optional
    MPEG4ExtensionDescriptorsBox (); // optional
    extra_boxes                    boxes;      // optional
}

// Use this if track is not HEVC compatible
class LHEVCSampleEntry() extends VisualSampleEntry ('lhv1', or 'lhe1') {
    LHVCConfigurationBox       lhvcconfig;

    MPEG4BitRateBox ();                        // optional
    MPEG4ExtensionDescriptorsBox (); // optional
    Box extra_boxes[];
}


aligned(8) class LHEVCDecoderConfigurationRecord {
    unsigned int(8) configurationVersion = 1;
    unsigned int(2) general_profile_space;
    unsigned int(1) general_tier_flag;
    unsigned int(5) general_profile_idc;
    unsigned int(32) general_profile_compatibility_flags;
    unsigned int(48) general_constraint_indicator_flags;
    unsigned int(8) general_level_idc;
    bit(1) complete_representation;
    bit(3) reserved = '111'b;
    unsigned int(12) min_spatial_segmentation_idc;
    bit(6) reserved = '111111'b;
    unsigned int(2) parallelismType;
    bit(6) reserved = '111111'b;
    unsigned int(2) chromaFormat;
    bit(5) reserved = '11111'b;
    unsigned int(3) bitDepthLumaMinus8;
    bit(5) reserved = '11111'b;
    unsigned int(3) bitDepthChromaMinus8;
    bit(16) avgFrameRate;
    bit(2) constantFrameRate;
```

```
    bit(3) numTemporalLayers;
    bit(1) temporalIdNested;
    unsigned int(2) lengthSizeMinusOne;
    unsigned int(8) numOfArrays;
    for (j=0; j < numOfArrays; j++) {
        bit(1) array_completeness;
        unsigned int(1) reserved = 0;
        unsigned int(6) NAL_unit_type;
        unsigned int(16) numNalus;
        for (i=0; i< numNalus; i++) {
            unsigned int(16) nalUnitLength;
            bit(8*nalUnitLength) nalUnit;
        }
    }
    unsigned int(16) operationPointIdx;
}


class TrackContentsInfoBox extends FullBox('tcon', version = 0, 0)){
    unsigned int (2) reserved
    unsigned int (6) num_layers_in_track
    for (i=0; i<num_layers_in_track; i++){
        unsigned int (4) reserved
        unsigned int (6) layer_id
        unsigned int (3) min_sub_layer_id
        unsigned int (3) max_sub_layer_id
    }
}


class OperationPointsInformation extends FullBox('oinf', version = 0, 0){
    unsigned int(16) scalability_mask
    unsigned int(2) reserved
    unsigned int(6) num_profile_tier_level
    for (i=1; i<=num_profile_tier_level; i++) {
        unsigned int(2) general_profile_space;
        unsigned int(1) general_tier_flag;
        unsigned int(5) general_profile_idc;
        unsigned int(32) general_profile_compatibility_flags;
        unsigned int(48) general_constraint_indicator_flags;
        unsigned int(8) general_level_idc;
    }
    unsigned int(16) num_operation_points
    for (i=0; i<num_operation_points) {
        unsigned int(16) operation_point_id
        unsigned int(8) max_temporal_id;
        unsigned int(8) layer_count
        for (i=0; i<layer_count; i++) {
            unsigned int(8) ptl_idx
            unsigned int(6) layer_id;
```

```
            unsigned int(1) is_outputlayer;
            unsigned int(1) is_alternate_outputlayer;
        }
    }
    unsigned int(8) max_layer_count
    for (i=0; i<max_layer_count; i++) {
        unsigned int(8) dependent_layerID
        unsigned int(8) num_layers_dependent_on
        for (j=0; j< num_layers_dependent_on; j++) {
            unsigned int(8) dependent_on_layerID
        }
        for (j = 0; j < 16; j++) {
            if (scalability mask & (1 << j))
                unsigned int(8) dimension_identifier
        }
    }
}
```

Basándose en la estructura actual de las cajas de arriba, y la información contenida dentro de ellas, para reproducir el contenido en el archivo, un reproductor puede estar configurado para primero encontrar la caja 'oinf (solo una en el archivo) para saber qué puntos de operación están incluidos, y luego elegir uno de los puntos de operación a ser reproducido. El reproductor de vídeo puede luego comprobar las cajas 'tcon' (una en cada pista que contiene vídeo L-HEVC) para saber qué pistas contienen las capas de los puntos de operación elegidos.

Con el uso básico de arriba del diseño actual en mente, esta divulgación propone que más información, tal como un formato de representación (el cual incluye resolución espacial, profundidad de bit, y formato de color), tasa de bit, y tasa de fotograma, sea incluida dentro de la caja 'oinf para habilitar la elección de puntos de operación. La entrada de muestra en cada pista no incluye un conjunto de tal información, sino solo para un punto de operación particular. Cuando múltiples puntos de operación están contenidos en una pista, la información para otros puntos de operación está ausente.

Otro problema se refiere al hecho de que la semántica de muchos de los campos en LHEVCDecoderConfigurationRecord no están claros, y algunos de ellos son confusos. Por ejemplo, el perfil, la capa y el nivel (PTL), chromaFormat, bitDepthLumaMinus8, y bitDepthChromaMinus8 son propiedades específicas de capa, pero actualmente se dice que se aplican al punto de operación indicado por operationPointIdx. Cuando el punto de operación contiene más de una capa, la semántica simplemente no es clara.

En realidad, basándose en los pasos del uso básico convencional del diseño, alguna de la información en la entrada de muestra no es realmente útil, particularmente cuando hay suficiente información en la caja 'oinf para la selección del punto de operación.

Pero otro problema es que, en SHVC y MV-HEVC, PTL solo es señalado para cada capa necesaria (esto es, una capa que es o bien una capa de salida o una capa a la que se hace referencia directamente o indirectamente por parte de una capa de salida dentro de un punto de operación o ambas), y no para cualquier capa no necesaria (una capa que no es una capa necesaria). Por lo tanto en el diseño de formato de archivo, puede no ser necesario señalar PTL para capas no necesarias.

Un resumen de los métodos y las técnicas descritos en esta divulgación está listado abajo. Implementaciones detalladas de ejemplo están proporcionadas en secciones posteriores. Los métodos y las técnicas de esta divulgación pueden ser aplicados independientemente o pueden ser aplicados en combinación.

Una primera técnica de esta divulgación incluye eliminar la señalización de la MPEG4BitRateBox() tras la LHEVCConfigurationBox dentro de la entrada de muestra LHEVC y la entrada de muestra HEVCLHVC. En cambio, habilitar la señalización de la información de tasa de bits para cada punto de operación en la caja 'oinf.

Una segunda técnica de esta divulgación incluye señalizar información sobre el formato de representación (lo cual incluye resolución espacial, profundidad de bit, y formato de color) para cada punto de punto de operación en la caja 'oinf.

Una tercera técnica de esta divulgación incluye eliminar del LHEVCDecoderConfigurationRecord la información PTL, la información de formato de representación, y la información de tasa de fotograma, las cuales están o bien ya proporcionadas en la caja ‘oinf o están propuestas para ser añadidas a la caja ‘oinf. La información restante en el LHEVCDecoderConfigurationRecord se aplica a todas las capas contenidas en la pista. En otro ejemplo de la tercera técnica, el diseño de LHEVCDecoderConfigurationRecord está restructurado de tal manera que la información de formato de representación y la información de tasa de fotograma, y posiblemente parámetros/información adicionales (p. ej., información de paralelismo), son señaladas para cada capa. El elemento de sintaxis unsigned int(2) parallelismType en un LHEVCDecoderConfigurationRecord puede indicar qué tipo de característica(s) de decodificación paralela puede ser usada para decodificar la imagen en la capa. Mosaico, frentes de onda, y segmentos son ejemplos de mecanismos de segmentación de imagen que pueden ser usados para facilitar para el procesamiento paralelo.

Una cuarta técnica de esta divulgación incluye eliminar el operationPointIdx del LHEVCDecoderConfigurationRecord. En otro ejemplo de la cuarta técnica, la señalización de una lista de índices de punto de operación que están asociados con la pista en el LHEVCDecoderConfigurationRecord está habilitada.

Una quinta técnica de esta divulgación incluye cambiar la semántica del campo layer_count en la caja ‘oinf para contar solo capas necesarias de un punto de operación.

Implementaciones de ejemplo de los métodos y las técnicas de la divulgación están descritos abajo. En los ejemplos de abajo, están mostrados cambios de texto relativos al formato de archivo HEVC y LHEVC. El texto añadido está mostrado entre los identificadores [START INSERTION] y [END INSERTION]. El texto eliminado está mostrado entre los identificadores [START DELETION] y [END DELETION].

Una primera implementación está descrita abajo.

Esta sección describe las modificaciones en detalle a la señalización de LHEVCSampleEntry, HEVCLHVCSampleEntry, LHVCDecoderConfigurationRecord y OperationPointsInformationBox (‘oinf) para las técnicas de la divulgación 1, 2, 3 (sin incluir su ejemplo a.), 4 (sin incluir su ejemplo a.) y 5.

```
class LHEVCConfigurationBox extends Box('lhvC') {
    LHEVCDecoderConfigurationRecord() LHEVCConfig;
}
class HEVCLHVCSampleEntry() extends HEVCSampleEntry() {
    LHEVCConfigurationBox          lhvcconfig;
    [START DELETION] MPEG4BitRateBox ();                                    //
optional [END DELETION]
    MPEG4ExtensionDescriptorsBox (); // optional
    extra_boxes                    boxes;      // optional
}
// Use this if track is not HEVC compatible
class LHEVCSampleEntry() extends VisualSampleEntry ('lhv1', or 'lhe1') {
    LHVCConfigurationBox       lhvcconfig;
    [START DELETION] MPEG4BitRateBox ();                                    //
optional [END DELETION]
    MPEG4ExtensionDescriptorsBox (); // optional
    Box extra_boxes[];
}
aligned(8) class LHEVCDecoderConfigurationRecord {
    unsigned int(8) configurationVersion = 1;
    [START DELETION] unsigned int(2) general_profile_space;
    unsigned int(1) general_tier_flag;
```

```
    unsigned int(5) general_profile_idc;
    unsigned int(32) general_profile_compatibility_flags;
    unsigned int(48) general_constraint_indicator_flags;
    unsigned int(8) general_level_idc; [END DELETION]
    bit(1) complete_representation;
    bit(3) reserved = '111'b;
    unsigned int(12) min_spatial_segmentation_idc;
    bit(6) reserved = '111111'b;
    unsigned int(2) parallelismType;
    [START DELETION] bit(6) reserved = '111111'b;
    unsigned int(2) chromaFormat;
    bit(5) reserved = '11111'b;
    unsigned int(3) bitDepthLumaMinus8;
    bit(5) reserved = '11111'b;
    unsigned int(3) bitDepthChromaMinus8;
    bit(16) avgFrameRate;
    bit(2) constantFrameRate; [END DELETION]
    [START INSERTION] bit(2) reserved = '11'b;  [END INSERTION]
    bit(3) numTemporalLayers;
    bit(1) temporalIdNested;
    unsigned int(2) lengthSizeMinusOne;
    unsigned int(8) numOfArrays;
    for (j=0; j < numOfArrays; j++) {
            bit(1) array_completeness;
            unsigned int(1) reserved = 0;
            unsigned int(6) NAL_unit_type;
            unsigned int(16) numNalus;
            for (i=0; i< numNalus; i++) {
                unsigned int(16) nalUnitLength;
                bit(8*nalUnitLength) nalUnit;
            }
    }
    [START DELETION] unsigned int(16) operationPointIdx; [END DELETION]
}
```

```
class OperationPointsInformation extends FullBox('oinf', version = 0, 0){
        unsigned int(16) scalability_mask
        unsigned int(2) reserved
        unsigned int(6) num_profile_tier_level
        for (i=1; i<=num_profile_tier_level; i++) {
                unsigned int(2) general_profile_space;
                unsigned int(1) general_tier_flag;
                unsigned int(5) general_profile_idc;
                unsigned int(32) general_profile_compatibility_flags;
                unsigned int(48) general_constraint_indicator_flags;
                unsigned int(8) general_level_idc;
        }
        unsigned int(16) num_operation_points
        for (i=0; i<num_operation_points) {
                unsigned int(16) operation_point_id
                unsigned int(8) max_temporal_id;
                unsigned int(8) layer_count;
                for (i=0; i<layer_count; i++) {
                        unsigned int(8) ptl_idx
                        unsigned int(6) layer_id;
                        unsigned int(1) is_outputlayer;
                        unsigned int(1) is_alternate_outputlayer;
                }
                [START INSERTION]
                unsigned int(16) minPicWidth;
                unsigned int(16) minPicHeight;
                unsigned int(16) maxPicWidth;
                unsigned int(16) maxPicHeight;
                unsigned int(2) maxChromaFormat;
                unsigned int(3) maxBitDepthMinus8;
                unsigned int(1) reserved
                unsigned int(1) frame_rate_info_flag
                unsigned int(1) bit_rate_info_flag
                if (frame_rate_info_flag) {
```

```
                bit(16) avgFrameRate;
                unsigned int(6) reserved
                bit(2) constantFrameRate;
            }
            if (bit_rate_info_flag) {
                unsigned int(32) maxBitRate;
                unsigned int(32) avgBitRate;
            }[END INSERTION]
        }
        unsigned int(8) max_layer_count
        for (i=0; i<max_layer_count; i++) {
            unsigned int(8) dependent_layerID
            unsigned int(8) num_layers_dependent_on
            for (j=0; j< num_layers_dependent_on; j++) {
                unsigned int(8) dependent_on_layerID
            }
            for (j = 0; j < 16; j++) {
                if (scalability mask & (1 << j))
                    unsigned int(8) dimension_identifier
            }
        }
    }
    ...
```

layer_count: este campo indica el número de capas [START INSERTION] necesarias [END INSERTION] que son una parte de [START INSERTION] el [END INSERTION] [START DELETION] un [END DELETION] punto de operación.

[START INSERTION]

minPicWidth especifica el valor mínimo de los indicadores de ancho luma como están definidos por el parámetro pic_width_in_luma_samples en ISO/IEC 23008-2 para el flujo del punto de operación.
minPicHeight especifica el valor mínimo de los indicadores de alto luma como están definidos por el parámetro pic_height_in_luma_samples en ISO/IEC 23008-2 para el flujo del punto de operación.
maxPicWidth especifica el valor máximo de los indicadores de ancho luma como están definidos por el parámetro pic_width_in_luma_samples en ISO/IEC 23008-2 para el flujo del punto de operación.
maxPicHeight especifica el valor máximo de los indicadores de alto luma como están definidos por el parámetro pic_height_in_luma_samples en ISO/IEC 23008-2 para el flujo del punto de operación.
maxChromaFormat especifica el valor máximo del indicador chroma_format como está definido por el parámetro chroma_format_idc en ISO/IEC 23008-2 para el flujo del punto de operación.
maxBitDepthMinus8 especifica el valor máximo de los indicadores de profundidad de bit luma y croma como están definidos por los parámetros bit_depth_luma_minus8 y bit_depth_chroma_minus8, respectivamente, en ISO/IEC 23008-2 para el flujo del punto de operación.

frame_rate_info_flag igual a 0 indica que ninguna información de tasa de fotograma está presente para el punto de operación. El valor 1 indica que la información de tasa de fotograma está presente para el punto de operación.
bit_rate_info_flag igual a 0 indica que ninguna información de tasa de bits está presente para el punto de operación. El valor 1 indica que la información de tasa de bits está presente para el punto de operación.
avgFrameRate da la tasa de fotograma promedio en unidades de fotogramas/(256 segundos) para el punto de operación. El valor 0 indica una tasa de fotograma promedio no especificada.
constantFrameRate igual a 1 indica que el flujo del punto de operación es de tasa de fotograma constante. El valor 2 indica que la representación de cada capa temporal en el flujo del punto de operación es de tasa de fotograma constante. El valor 0 indica que el flujo del punto de operación puede o no ser de tasa de fotograma constante.
maxBitRate da la tasa de bits máxima en bits/segundo del flujo del punto de operación, a lo largo de cualquier ventana de un segundo.
avgBitRate da la tasa de bits promedio en bits/segundo del flujo del punto de operación.

[END INSERTION]

Una segunda implementación está descrita abajo.

Esta sección describe las modificaciones en detalle a la señalización de LHVCDecoderConfigurationRecord para el ejemplo de la divulgación 3(a).

```
aligned(8) class LHEVCDecoderConfigurationRecord {
    unsigned int(8) configurationVersion = 1;
    [START DELETION] unsigned int(2) general_profile_space;
    unsigned int(1) general_tier_flag;
    unsigned int(5) general_profile_idc;
    unsigned int(32) general_profile_compatibility_flags;
    unsigned int(48) general_constraint_indicator_flags;
    unsigned int(8) general_level_idc;
    bit(1) complete_representation;
    bit(3) reserved = '111'b; [END DELETION]
    [START INSERTION] bit(2) reserved = '11'b; [END INSERTION]
    [START INSERTION] unsigned int(6) num_layers; [END INSERTION]
    for (j=0; j < num_layers; j++) {
        [START INSERTION] unsigned int(8) layer_id; [END INSERTION]
        unsigned int(12) min_spatial_segmentation_idc;
        bit(6) reserved = '111111'b;
        unsigned int(2) parallelismType;
        bit(6) reserved = '111111'b;
        unsigned int(2) chromaFormat;
        [START INSERTION] bit(6) reserved = '111111'b; [END INSERTION]
        [START DELETION] bit(5) reserved = '11111'b; [ENDDELETION]
        unsigned int(3) bitDepthLumaMinus8;
        bit(5) reserved = '11111'b;
        unsigned int(3) bitDepthChromaMinus8;
        [START INSERTION] bit(5) reserved = '11111'b; [END INSERTION]
    [START DELETION] bit(16) avgFrameRate;
    bit(2) constantFrameRate; [END DELETION]
        bit(3) numTemporalLayers;
        bit(1) temporalIdNested;
        [START INSERTION] bit(4) reserved = '1111'b; [END INSERTION]
    }
    [START INSERTION] bit(1) complete_representation; [END INSERTION]
```

```
        unsigned int(2) lengthSizeMinusOne;
        [START INSERTION] bit(5) reserved = '11111'b; [END INSERTION]
        unsigned int(8) numOfArrays;
        for (j=0; j < numOfArrays; j++) {
                bit(1) array_completeness;
                unsigned int(1) reserved = 0;
                unsigned int(6) NAL_unit_type;
                unsigned int(16) numNalus;
                for (i=0; i< numNalus; i++) {
                    unsigned int(16) nalUnitLength;
                    bit(8*nalUnitLength) nalUnit;
                }
        }
        [START DELETION] unsigned int(16) operationPointIdx; [END DELETION]
}
[START INSERTION]
num_layers specifies the number of layers in the track.
layer_id specifies the layer ID value for which the information in this loop is provided.
[END INSERTION]
```

Una tercera implementación está descrita abajo.

Esta sección describe las modificaciones en detalle a la señalización de LHVCDecoderConfigurationRecord para el ejemplo de la divulgación 4(a).

```
aligned(8) class LHEVCDecoderConfigurationRecord {
    unsigned int(8) configurationVersion = 1;
    unsigned int(2) general_profile_space;
    unsigned int(1) general_tier_flag;
    unsigned int(5) general_profile_idc;
    unsigned int(32) general_profile_compatibility_flags;
    unsigned int(48) general_constraint_indicator_flags;
    unsigned int(8) general_level_idc;
    bit(1) complete_representation;   bit(3) reserved = '111'b;
    unsigned int(12) min_spatial_segmentation_idc;
    bit(6) reserved = '111111'b;
```

```
        unsigned int(2) parallelismType;
        bit(6) reserved = '111111'b;
        unsigned int(2) chromaFormat;
        bit(5) reserved = '11111'b;
        unsigned int(3) bitDepthLumaMinus8;
        bit(5) reserved = '11111'b;
        unsigned int(3) bitDepthChromaMinus8;
        bit(16) avgFrameRate;
        bit(2) constantFrameRate;
        bit(3) numTemporalLayers;
        bit(1) temporalIdNested;
        unsigned int(2) lengthSizeMinusOne;
        unsigned int(8) numOfArrays;
        for (j=0; j < numOfArrays; j++) {
                bit(1) array_completeness;
                unsigned int(1) reserved = 0;
                unsigned int(6) NAL_unit_type;
                unsigned int(16) numNalus;
                for (i=0; i< numNalus; i++) {
                    unsigned int(16) nalUnitLength;
                    bit(8*nalUnitLength) nalUnit;
                }
        }
        [START DELETION] unsigned int(16) operationPointIdx; [END DELETION]
        [START INSERTION]
        unsigned int(16) numOfOperationPoints;
        for (j=0; j < numOfOperationPoints; j++) {
                unsigned int(16) operationPointIdx;
        } [END INSERTION]
}
```

[START INSERTION] numOperationPoints: este campo señala el número de puntos de operación que están disponibles para la pista. [END INSERTION]

operationPointIdx: este campo señala el índice del punto de operación documentado en la caja de información de punto de operación. [START DELETION] Los valores de general_profile_space, general tier flag, general_profile_idc, general_profile_compatibility flags, general_constraint_indicator_flag y general_level_idc en LHEVCDecoderConfigurationRecord serán los mismos como los valores respectivos del punto de operación operationPointIdx-ésimo en la caja de información de punto de operación. [END DELETION] [START INSERTION] El

valor de max_temporal_id en el punto de operación operationPointIdx-ésimo en la caja de información de punto de operación será menor que o igual al valor de numTemporalLayers. [END INSERTION]

NOTA Una pista puede estar asociada con un o [START DELETION] representar [END DELETION] más de un conjunto de capas de salida [START DELETION] y por consiguiente más de un perfil [START DELETION]. Un reproductor puede averiguar qué capas deben ser decodificadas y qué capas deben ser transmitidas conforme a la información de perfil en LHEVCDecoderConfigurationRecord [START INSERTION] para el punto de operación seleccionado con operationIndexIdx de índice [END INSERTION] investigando la información proporcionada para el punto de operación operationPointIdx-ésimo en la caja de información de punto de operación.

NOTA Para cada capa de imagen auxiliar incluida en la pista, es recomendable incluir, dentro de nalUnit, una unidad NAL SEI que contiene un mensaje SEI declarativo, tal como el mensaje SEI de información de representación de profundidad para capas de imagen auxiliares de profundidad, que especifican características de la capa de imagen auxiliar.

La FIG. 2 es un diagrama de flujos que ilustra un codificador de vídeo 20 de ejemplo que puede implementar las técnicas descritas en esta divulgación. El codificador de vídeo 20 puede estar configurado para transmitir datos de vídeo de vista única, multivista, escalable, 3D, y de otros tipos. El codificador de vídeo 20 puede estar configurado para transmitir vídeo a la entidad de posprocesamiento 27. La entidad de posprocesamiento 27 está destinada a representar un ejemplo de una entidad de vídeo, tal como un MANE o dispositivo de empalme/editado, que puede procesar datos de vídeo codificados del codificador de vídeo 20. En algunos ejemplos, la entidad de procesamiento de posprocesamiento puede ser un ejemplo de una entidad de red. En algunos sistemas de codificación de vídeo, la entidad de posprocesamiento 27 y el codificador de vídeo 20 pueden ser partes de dispositivos separados, mientras que en otros ejemplos, la funcionalidad descrita con respecto a la entidad de posprocesamiento 27 puede ser realizada por el mismo dispositivo que comprende codificador de vídeo 20. La entidad de posprocesamiento 27 puede ser un dispositivo de vídeo. En algunos ejemplos, la entidad de posprocesamiento 27 puede ser la misma como el dispositivo de generación de archivos 34 de la FIG. 1.

El codificador de vídeo 20 puede realizar intra e intercodificación de bloques de vídeo dentro de segmentos de vídeo. La intracodificación se basa en la predicción espacial para reducir o eliminar la redundancia espacial en vídeo dentro de un fotograma o imagen de vídeo dados. La intercodificación se basa en la predicción temporal para reducir o eliminar la redundancia temporal en vídeo dentro de segmentos o imágenes adyacentes de una secuencia de vídeo. El modo intra (modo I) se puede referir a cualquiera de varios modos de compresión con base espacial. Los modos inter, tales como la predicción unidireccional (modo P) o la bipredicción (modo B), se puede referir a cualquiera de varios modos de compresión con base temporal.

En el ejemplo de la FIG. 2, el codificador de vídeo 20 incluye una unidad de partición 37, unidad de procesamiento de predicción 41, unidad de filtro 63, memoria de imágenes de referencia 64, sumador 50, unidad de procesamiento de transformación 52, unidad de cuantización 54, y unidad de codificación por entropía 56. La unidad de procesamiento de predicción 41 incluye unidad de estimación de movimiento 42, unidad de compensación de movimiento 44, y unidad de procesamiento de intrapredicción 46. Para la reconstrucción de bloque de vídeo, el codificador de vídeo 20 también incluye unidad de cuantización inversa 58, unidad de procesamiento de transformación inversa 60, y sumador 62. La unidad de filtro 63 está destinada a representar uno o más filtros de bucle tales como un filtro de desbloqueo, un filtro de bucle adaptativo (ALF), y un filtro de desplazamiento adaptativo de muestras (SAO). Aunque la unidad de filtro 63 está mostrada en la FIG. 2 como que es un filtro de bucle, en otras configuraciones, la unidad de filtro 63 puede ser implementada como un filtro posbucle.

La memoria de datos de vídeo 35 del codificador de vídeo 20 puede almacenar datos de vídeo a ser codificados por los componentes del codificador de vídeo 20. Los datos de vídeo almacenados en la memoria de datos de vídeo 35 pueden ser obtenidos, por ejemplo, de la fuente de vídeo 18. La memoria de imágenes de referencia 64 puede ser una memoria de imágenes de referencia que almacena datos de vídeo de referencia para el uso en la codificación de datos de vídeo por parte del codificador de vídeo 20, p. ej., en modos de intra o intercodificación. La memoria de datos de vídeo 35 y la memoria de imágenes de referencia 64 pueden estar formadas por cualquiera de una variedad de dispositivos de memoria, tales como memoria dinámica de acceso aleatorio (DRAM), incluido DRAM síncrona (SDRAM), RAM magnetorresistiva (MRAM), RAM resistiva (RRAM), u otros tipos de dispositivos de memoria. La memoria de datos de vídeo 35 y la memoria de imágenes de referencia 64 pueden estar proporcionadas por el mismo dispositivo de memoria o dispositivos de memoria separados. En varios ejemplos, la memoria de datos de vídeo 35 puede ser en chip con otros componentes del codificador de vídeo 20, o fuera de chip en relación con esos componentes.

Como está mostrado en la FIG. 2, el codificador de vídeo 20 recibe datos de vídeo, y la unidad de partición 37 particiona los datos en bloques de vídeo. Esta partición también puede incluir la partición en segmentos, mosaicos, u otras unidades más grandes, así como partición de bloque de vídeo, p. ej., de acuerdo con una estructura de árbol cuaternario de LCU y CU. El codificador de vídeo 20 ilustra generalmente los componentes que codifican bloques de vídeo dentro de un segmento de vídeo a ser codificado. El segmento puede ser dividido en múltiples bloques de vídeo (y posiblemente en conjuntos de bloques de vídeo referidos como mosaicos). La unidad de procesamiento de

predicción 41 puede seleccionar uno de una pluralidad de modos de codificación posibles, tales como uno de una pluralidad de modos de intracodificación o uno de una pluralidad de modos de intercodificación, para el bloque de vídeo actual basado en resultados de error (p. ej., tasa de codificación y el nivel de distorsión). La unidad de procesamiento de predicción 41 puede proporcionar el bloque intra o intercodificado resultante al sumador 50 para generar datos de bloque residuales y al sumador 62 para reconstruir el bloque codificado para el uso como una imagen de referencia.

La unidad de procesamiento de intrapredicción 46 dentro de la unidad de procesamiento de predicción 41 puede realizar codificación intrapredictiva del bloque de vídeo actual en relación con uno o más bloques vecinos en el mismo fotograma o segmento como el bloque actual a ser codificado para proporcionar compresión espacial. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 dentro de la unidad de procesamiento de predicción 41 realizan codificación interpredictiva del bloque de vídeo actual en relación con uno o más bloques predictivos en una o más imágenes de referencia para proporcionar compresión temporal.

La unidad de estimación de movimiento 42 puede estar configurada para determinar el modo de interpredicción para un segmento de vídeo de acuerdo con un patrón predeterminado para una secuencia de vídeo. El patrón predeterminado puede designar segmentos de vídeo en la secuencia como segmentos P, segmentos B o segmentos GPB. La unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44 pueden estar altamente integradas, pero están ilustradas separadamente por razones conceptuales. La estimación de movimiento, realizada por la unidad de estimación de movimiento 42, es el proceso de generar 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 un fotograma o imagen de vídeo actual en relación con un bloque predictivo dentro de una imagen de referencia.

Un bloque predictivo es un bloque que resulta que se corresponde estrechamente con la PU del bloque de vídeo a ser codificado en términos de diferencia de píxel, lo cual puede ser determinado mediante suma de diferencia absoluta (SAD), suma de diferencia al cuadrado (SSD), u otras métricas de diferencia. En algunos ejemplos, el codificador de vídeo 20 puede calcular valores para posiciones de píxeles de subentero 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 de píxel fraccionales de la imagen de referencia. Por lo tanto, la unidad de estimación de movimiento 42 puede realizar una búsqueda de movimiento en relación con las posiciones de píxel completo y posiciones de píxel fraccionales y transmitir un vector de movimiento con precisión de píxel fraccional.

La unidad de estimación de movimiento 42 calcula un vector de movimiento para una PU de un bloque de vídeo en un segmento intercodificado comparando la posición de la PU con la posición de un bloque predictivo de una imagen de referencia. La imagen de referencia puede ser seleccionada de una primera lista de imágenes de referencia (Lista 0) o una segunda lista de imágenes de referencia (Lista 1), cada una de las cuales identifican 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 la unidad de compensación de movimiento 44.

La compensación de movimiento, realizada por la unidad de compensación de movimiento 44, puede implicar extraer o generar el bloque predictivo basándose en el vector de movimiento determinado mediante estimación de movimiento, posiblemente realizando interpolaciones a precisión de subpíxel. Tras 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 cual apunta el vector de movimiento en una de las listas de imágenes de referencia. El codificador de vídeo 20 puede formar un bloque de vídeo residual restando valores de píxel del bloque predictivo a los valores de píxel del bloque de vídeo actual que es codificado, formando valores de diferencia de píxel. Los valores de diferencia de píxel forman valores residuales para el bloque y pueden incluir componentes de diferencia tanto luma como croma. El sumador 50 representa el componente o los componentes que realizan esta operación de resta. La unidad de compensación de movimiento 44 también puede generar elementos de sintaxis asociados con los bloques de vídeo y el segmento de vídeo para el uso por parte del decodificador de vídeo 30 en la decodificación de los bloques de vídeo del segmento de vídeo.

La unidad de procesamiento de intrapredicción 46 puede intrapredecir un bloque actual, como una alternativa a la interpredicción realizada por la unidad de estimación de movimiento 42 y la unidad de compensación de movimiento 44, como está descrito arriba. En particular, la unidad de procesamiento de intrapredicción 46 puede determinar un modo de intrapredicción a usarse para codificar un bloque actual. En algunos ejemplos, la unidad de procesamiento de intrapredicción 46 puede codificar un bloque actual usando varios modos de intrapredicción, p. ej., durante pases de codificación separados, y la unidad de procesamiento de intrapredicción 46 (o unidad de selección de modo 40, en algunos ejemplos) puede seleccionar un modo de intrapredicción apropiado a usarse de los modos probados. Por ejemplo, la unidad de procesamiento de intrapredicción 46 puede calcular valores de tasa-distorsión usando un análisis de tasa-distorsión para los varios modos de intrapredicción probados, y seleccionar el modo de intrapredicción que tiene las mejores características de tasa-distorsión de entre los modos probados. El análisis de tasa-distorsión determina generalmente una cantidad de distorsión (o error) entre un bloque codificado y un bloque no codificado

original que fue codificado para producir el bloque codificado, así como una tasa de bits (esto es, un número de bits) usada para producir el bloque codificado. La unidad de procesamiento de intrapredicción 46 puede calcular ratios a partir de las distorsiones y tasas para los varios bloques codificados para determinar qué modo de intrapredicción exhibe el mejor valor de tasa-distorsión para el bloque.

En cualquier caso, tras seleccionar un modo de intrapredicción para un bloque, la unidad de procesamiento de intrapredicción 46 puede proporcionar información indicativa del modo de intrapredicción seleccionado para el bloque a la unidad de codificación por entropía 56. La unidad de codificación por entropía 56 puede codificar la información que indica el modo de intrapredicción seleccionado de conformidad con las técnicas de esta divulgación. El codificador de vídeo 20 puede incluir en el flujo de bits transmitido datos de configuración, los cuales pueden incluir una pluralidad de tablas de índices de modo de intrapredicción y una pluralidad de tablas de índices de modo de intrapredicción modificadas (también referidas como tablas de asignación de palabras de código), definiciones de contextos de codificación para varios bloques, e indicaciones de un modo de intrapredicción sumamente probable, una tabla de índices de modo de intrapredicción, y una tabla de índices de modo de intrapredicción modificada a usarse para cada uno de los contextos.

Después de que la unidad de procesamiento de predicción 41 genera el bloque predictivo para el bloque de vídeo actual a través de o bien interpredicción o intrapredicción, el codificador de vídeo 20 puede formar un bloque de vídeo residual restando el bloque predictivo del bloque de vídeo actual. Los datos de vídeo residuales en el bloque residual pueden estar incluidos en una o más TU y ser aplicados a la unidad de procesamiento de transformación 52. La unidad de procesamiento de transformación 52 transforma los datos de vídeo residuales en coeficientes de transformación residuales usando una transformación, tal como una transformación de coseno discreta (DCT) o una transformación conceptualmente similar. La unidad de procesamiento de transformación 52 puede convertir los datos de vídeo residuales de un dominio 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 más la tasa de bits. El proceso de cuantización puede reducir la profundidad de bit asociada con algunos de o todos los coeficientes. El grado de la cuantización puede ser modificado ajustando un parámetro de cuantización. En algunos ejemplos, la unidad de cuantización 54 puede realizar luego un escaneo de la matriz incluyendo los coeficientes de transformación cuantizados. Alternativamente, la unidad de codificación por entropía 56 puede realizar el escaneo.

Después de la cuantización, la unidad de codificación por entropía 56 puede codificar por entropía elementos de sintaxis que representan los coeficientes de transformación cuantizados. Por ejemplo, la unidad de codificación por entropía 56 puede realizar codificación adaptativa según el contexto de longitud variable (CAVLC), codificación aritmética binaria adaptativa al contexto (CABAC), codificación aritmética binaria adaptativa al contexto basada en sintaxis (SBAC), codificación por entropía de partición de intervalos de probabilidad (PIPE) u otra metodología o técnica de codificación por entropía. Después de la codificación por entropía por la unidad de codificación por entropía 56, el flujo de bits codificado puede ser transmitido al decodificador de vídeo 30, o archivado para la posterior transmisión o recuperación por el decodificador de vídeo 30. La unidad de codificación por entropía 56 también puede codificar por entropía los vectores de movimiento y los otros elementos de sintaxis para el segmento de vídeo actual que es codificado.

La unidad de cuantización inversa 58 y la unidad de procesamiento de transformación inversa 60 aplican cuantización inversa y transformación inversa, respectivamente, para reconstruir el bloque residual en el dominio de píxel para el uso posterior como un bloque de referencia de una imagen de referencia. La unidad de compensación de movimiento 44 puede calcular un bloque de referencia añadiendo el bloque residual a un bloque predictivo de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. La unidad de compensación de movimiento 44 también puede aplicar uno o más filtros de interpolación al bloque residual reconstruido para calcular valores de píxeles de subentero para el uso en la estimación de movimiento. El sumador 62 añade el bloque residual reconstruido al bloque de predicción con compensación de movimiento producido por la unidad de compensación de movimiento 44 para producir un bloque de referencia para el almacenamiento en la memoria de imágenes de referencia 64. El bloque de referencia 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 interpredecir un bloque en un fotograma o imagen de vídeo subsiguiente.

El codificador de vídeo 20 representa un ejemplo de un codificador de vídeo configurado para generar datos de vídeo que pueden ser almacenados usando las técnicas de formato de archivo descritas en esta divulgación.

La FIG. 3 es un diagrama de bloques que ilustra un decodificador de vídeo 30 de ejemplo que puede implementar las técnicas descritas en esta divulgación. El decodificador de vídeo 30 puede estar configurado para decodificar datos de vídeo de vista única, multivista, escalable, 3D, y de otros tipos. En el ejemplo de la FIG. 3, el decodificador de vídeo 30 incluye una unidad de decodificación por entropía 80, unidad de procesamiento de predicción 81, unidad de cuantización inversa 86, unidad de procesamiento de transformación inversa 88, sumador 90, unidad de filtro 91, y memoria de imágenes de referencia 92. La unidad de procesamiento de predicción 81 incluye unidad de compensación de movimiento 82 y unidad de procesamiento de intrapredicción 84. 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 de la FIG. 2.

Un búfer de imagen codificado (CPB) 79 puede recibir y almacenar datos de vídeo codificados (p. ej., unidades NAL) de un flujo de bits. Los datos de vídeo almacenados en el CPB 79 pueden ser obtenidos, por ejemplo, a partir del enlace 16, p. ej., a partir de una fuente de vídeo local, tal como una cámara, a través de comunicación de red inalámbrica o por cable de datos de vídeo, o accediendo a medios de almacenamiento de datos físicos. El CPB 79 puede formar una memoria de datos de vídeo que almacena datos de vídeo codificados de un flujo de bits de vídeo codificado. El CPB 79 puede ser una memoria de imágenes de referencia que almacena datos de vídeo de referencia para el uso en la decodificación de datos de vídeo por parte del decodificador de vídeo 30, p. ej., en los modos de intra o intercodificación. El CPB 79 y la memoria de imágenes de referencia 92 pueden estar formadas por cualquiera de una variedad de dispositivos de memoria, tales como memoria dinámica de acceso aleatorio (DRAM), incluido DRAM síncrona (SDRAM), RAM magnetorresistiva (MRAM), RAM resistiva (RRAM), u otros tipos de dispositivos de memoria. El CPB 79 y la memoria de imágenes de referencia 92 pueden ser proporcionadas por el mismo dispositivo de memoria o dispositivos de memoria separados. En varios ejemplos, el CPB 79 puede ser en chip con otros componentes del decodificador de vídeo 30, o fuera de chip en relación con esos componentes.

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 segmento de vídeo codificado y elementos de sintaxis asociados del codificador de vídeo 20. El decodificador de vídeo 30 puede recibir el flujo de bits de vídeo codificado de la entidad de red 29. La entidad de red 29 puede, por ejemplo, ser un servidor, un MANE, un editor/empalmador de vídeo, u otro dispositivo tal configurado para implementar una o más de las técnicas descritas arriba. La entidad de red 29 puede o no incluir un codificador de vídeo, tal como el codificador de vídeo 20. Algunas de las técnicas descritas en esta divulgación pueden ser implementadas por la entidad de red 29 antes de que la entidad de red 29 transmita el flujo de bits de vídeo codificado al decodificador de vídeo 30. En algunos sistemas de decodificación de vídeo, la entidad de red 29 y el decodificador de vídeo 30 pueden ser partes de dispositivos separados, mientras que en otros ejemplos, la funcionalidad descrita con respecto a la entidad de red 29 puede ser realizada por el mismo dispositivo que comprende el decodificador de vídeo 30. Se puede considerar que la entidad de red 29 es un dispositivo de vídeo. Además, en algunos ejemplos, la entidad de red 29 es el dispositivo de generación de archivos 34 de la FIG. 1.

La unidad de decodificación por entropía 80 del decodificador de vídeo 30 decodifica por entropía elementos de sintaxis particulares del flujo de bits para generar coeficientes cuantizados, vectores de movimiento, y otros elementos de sintaxis. La unidad de decodificación por entropía 80 transmite los vectores de movimiento y otros elementos de sintaxis a la unidad de procesamiento de predicción 81. El decodificador de vídeo 30 puede recibir los elementos de sintaxis al nivel de segmento de vídeo y/o el nivel de bloque de vídeo.

Cuando el segmento de vídeo es codificado como un segmento intracodificado (I), la unidad de procesamiento de intrapredicción 84 de la unidad de procesamiento de predicción 81 puede generar datos de predicción para un bloque de vídeo del segmento de vídeo actual basándose en un modo de intrapredicción señalado y datos de bloques previamente decodificados del fotograma o imagen actual. Cuando el fotograma de vídeo es codificado como un segmento intercodificado (p. ej., B, P o GPB), la unidad de compensación de movimiento 82 de la unidad de procesamiento de predicción 81 produce bloques predictivos para un bloque de vídeo del segmento de vídeo actual basándose en los vectores de movimiento y otros elementos de sintaxis recibidos de la unidad de decodificación por entropía 80. Los bloques predictivos pueden ser producidos a partir de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. El decodificador de vídeo 30 puede construir las listas de fotogramas de referencia, Lista 0 y Lista 1, usando técnicas de construcción por defecto basándose en las imágenes de referencia almacenadas en la memoria de imágenes de referencia 92.

La unidad de compensación de movimiento 82 determina la información de predicción para un bloque de vídeo del segmento de vídeo actual analizando los vectores de movimiento y otros elementos de sintaxis, y usa la información de predicción para producir los bloques predictivos para el bloque de vídeo actual que es codificado. Por ejemplo, la unidad de compensación de movimiento 82 usa algunos de elementos de sintaxis recibidos para determinar un modo de predicción (p. ej., intra o interpredicción) usado para codificar los bloques de vídeo del segmento de vídeo, un tipo de segmento de interpredicción (p. ej., segmento B, segmento P, o segmento GPB), información de construcción para una o más de las listas de imágenes de referencia para el segmento, vectores de movimiento para cada bloque de vídeo intercodificado del segmento, estado de interpredicción para cada bloque de vídeo intercodificado del segmento, y otra información para decodificar los bloques de vídeo en el segmento de vídeo actual.

La unidad de compensación de movimiento 82 también puede realizar interpolación basándose en filtros de interpolación. La unidad de compensación de movimiento 82 puede usar filtros de interpolación como son usados por el codificador de vídeo 20 durante la codificación de los bloques de vídeo para calcular valores interpolados para píxeles de subentero de bloques de referencia. En este caso, la unidad de compensación de movimiento 82 puede determinar los filtros de interpolación usados por el codificador de vídeo 20 a partir de los elementos de sintaxis recibidos y puede usar los filtros de interpolación para producir bloques predictivos.

La unidad de cuantización inversa 86 cuantiza inversamente, esto es, decuantiza, los coeficientes de transformación cuantizados proporcionados en el flujo de bits y decodificados por la unidad de decodificación por entropía 80. El proceso de cuantización inversa puede incluir el uso de un parámetro de cuantización calculado por el codificador de vídeo 20 para cada bloque de vídeo en el segmento de vídeo para determinar un grado de cuantización y, de igual modo, un grado de cuantización inversa que debe aplicarse. La unidad de procesamiento de transformación inversa 88 aplica una transformación inversa, p. ej., una DCT inversa, una transformación de entero inversa, o un proceso de transformación inversa conceptualmente similar, a los coeficientes de transformación para producir bloques residuales en el dominio de píxel.

Después de que la unidad de compensación de movimiento 82 genera el bloque predictivo para el bloque de vídeo actual basándose en los vectores de movimiento y otros elementos de sintaxis, el decodificador de vídeo 30 forma un bloque de vídeo decodificado sumando los bloques residuales de la unidad de procesamiento de transformación inversa 88 con los bloques predictivos correspondientes generados por la unidad de compensación de movimiento 82. El sumador 90 representa el componente o componentes que realizan esta operación de suma. Si se desea, los filtros de bucle (o bien en el bucle de codificación o bien tras el bucle de codificación) también pueden ser usados para suavizar las transiciones de píxeles, o de otro modo mejorar la calidad del vídeo. La unidad de filtro 91 está destinada a representar uno o más filtros de bucle tal como un filtro de desbloqueo, un filtro de bucle adaptativo (ALF), y un filtro de desplazamiento adaptativo de muestras (SAO). Aunque la unidad de filtro 91 está mostrada en la FIG. 3 como que es un filtro en bucle, en otras configuraciones, la unidad de filtro 91 puede ser implementada como un filtro posbucle. Los bloques de vídeo decodificados en un fotograma o imagen dados son luego almacenados en la memoria de imágenes de referencia 92, la cual almacena imágenes de referencia usadas para la subsiguiente compensación de movimiento. La memoria de imágenes de referencia 92 también almacena vídeo decodificado para la posterior presentación en un dispositivo de pantalla, tal como el dispositivo de pantalla 32 de la FIG. 1.

El decodificador de vídeo 30 de la FIG. 3 representa un ejemplo de un decodificador de vídeo configurado para decodificar datos de vídeo que pueden ser almacenados usando las técnicas de formato de archivo descritas en esta divulgación.

La FIG. 4 es un diagrama de bloques que ilustra un conjunto de dispositivos de ejemplo que forman parte de una red 100. En este ejemplo, la red 100 incluye dispositivos de enrutamiento 104A, 104B (dispositivos de enrutamiento 104) y dispositivo de transcodificación 106. Los dispositivos de enrutamiento 104 y el dispositivo de transcodificación 106 están destinados a representar un pequeño número de dispositivos que pueden formar parte de la red 100. Otros dispositivos de red, tales como interruptores, concentradores, pasarelas, cortafuegos, puentes, y otros dispositivos tales también pueden estar incluidos dentro de la red 100. Además, dispositivos de red adicionales pueden ser proporcionados junto con una ruta de red entre el dispositivo servidor 102 y el dispositivo cliente 108. El dispositivo servidor 102 puede corresponder al dispositivo de origen 12 (FIG. 1), mientras que el dispositivo cliente 108 puede corresponder al dispositivo de destino 14 (FIG. 1), en algunos ejemplos.

En general, los dispositivos de enrutamiento 104 implementan uno o más protocolos de enrutamiento para intercambiar datos de red a través de la red 100. En algunos ejemplos, los dispositivos de enrutamiento 104 pueden estar configurados para realizar operaciones de proxy o de caché. Por lo tanto, en algunos ejemplos, a los dispositivos de enrutamiento 104 se puede hacer referencia como dispositivos proxy. En general, los dispositivos de enrutamiento 104 ejecutan protocolos de enrutamiento para descubrir rutas a través de la red 100. Al ejecutar tales protocolos de enrutamiento, el dispositivo de enrutamiento 104B puede descubrir una ruta de red desde sí mismo hasta el dispositivo servidor 102 a través del dispositivo de enrutamiento 104A.

Las técnicas de esta divulgación pueden ser implementadas por dispositivos de red tales como los dispositivos de enrutamiento 104 y el dispositivo de transcodificación 106, pero también pueden ser implementadas por el dispositivo cliente 108. De esta manera, los dispositivos de enrutamiento 104, el dispositivo de transcodificación 106, y el dispositivo cliente 108 representan ejemplos de dispositivos configurados para llevar a cabo las técnicas de esta divulgación. Además, los dispositivos de la FIG. 1, y el codificador 20 ilustrado en la FIG. 2 y el decodificador 30 ilustrado en la FIG. 3 son también ejemplos de dispositivos que pueden ser configurados para llevar a cabo una o más de las técnicas de esta divulgación.

La FIG. 5A es un diagrama conceptual que ilustra una estructura de ejemplo de un archivo 300, de conformidad con una o más de las técnicas de esta divulgación. En el ejemplo de la FIG. 5A, el archivo 300 incluye una caja de película 302 y una pluralidad de cajas de datos de medios 304. Aunque están ilustradas en el ejemplo de la FIG. 5A como que están en el mismo archivo, en otros ejemplos la caja de película 302 y las cajas de datos de medios 304 pueden estar en archivos separados. Como está indicado arriba, una caja puede ser un bloque de construcción orientado al objeto definido por un identificador de tipo único y una longitud. Por ejemplo, una caja puede ser la estructura de sintaxis elemental en el ISOBMFF, que incluye un tipo de caja codificada de cuatro caracteres, un recuento de bytes de la caja, y una carga útil.

La caja de película 302 puede contener metadatos para pistas del archivo 300. Cada pista del archivo 300 puede comprender un flujo continuo de datos de medios. Cada una de las cajas de datos de medios 304 puede incluir una o más muestras 305. Cada una de las muestras 305 puede comprender una unidad de acceso de audio o vídeo. Como

está descrito en otro lugar en esta divulgación, cada unidad de acceso puede comprender múltiples imágenes codificadas en codificación multivista (p. ej., MV-HEVC y 3D-HEVC) y codificación de vídeo escalable (p. ej., SHVC). Por ejemplo, una unidad de acceso puede incluir una o más imágenes codificadas para cada capa.

Además, en el ejemplo de la FIG. 5A, la caja de película 302 incluye una caja de pista 306. La caja de pista 306 puede encerrar metadatos para una pista del archivo 300. En otros ejemplos, la caja de película 302 puede incluir múltiples cajas de pista para diferentes pistas del archivo 300. La caja de pista 306 incluye una caja de medios 307. La caja de medios 307 puede contener todos los objetos que declaran información sobre los datos de medios dentro de la pista. La caja de medios 307 incluye una caja de información de medios 308. La caja de información de medios 308 puede contener todos los objetos que declaran información característica de los medios de la pista. La caja de información de medios 308 incluye una caja de tabla de muestras 309. La caja de tabla de muestras 309 puede especificar metadatos específicos de muestra.

En el ejemplo de la FIG. 5A, la caja de tabla de muestras 309 incluye una caja SampleToGroup 310 y una caja SampleGroupDescription 312, y la caja SampleGroupDescription 312 incluye la caja oinf 316. En otros ejemplos, la caja de tabla de muestras 309 puede incluir otras cajas además de la caja SampleToGroup 310 y la caja SampleGroupDescription 312, y/o puede incluir múltiples cajas SampleToGroup y cajas SampleGroupDescription. La caja SampleToGroup 310 puede asignar muestras (p. ej., unas particulares de las muestras 305) a un grupo de muestras. La caja SampleGroupDescription 312 puede especificar una propiedad compartida por las muestras en el grupo de muestras (esto es, grupo de muestra). Además, la caja de tabla de muestras 309 puede incluir una pluralidad de cajas de entrada de muestras 311. Cada una de las cajas de entrada de muestras 311 puede corresponder a una muestra en el grupo de muestras. En algunos ejemplos, las cajas de entrada de muestras 311 son instancias de una clase de entrada de muestra accesible aleatoriamente que amplía una clase de descripción de grupo de muestras base.

De conformidad con una o más técnicas de esta divulgación, la caja SampleGroupDescription 312 puede especificar que cada muestra del grupo de muestra contiene al menos una imagen IRAP. De esta forma, el dispositivo de generación de archivos 34 puede generar un archivo que comprende una caja de pista 306 que contiene metadatos para una pista en el archivo 300. Los datos de medios para la pista comprende una secuencia de muestras 305. Cada una de las muestras puede ser una unidad de acceso de vídeo de datos de vídeo multicapa (p. ej., datos de vídeo SHVC, MV-HEVC, o 3D-HEVC). Además, como parte de generar el archivo 300, el dispositivo de generación de archivos 34 puede generar, en el archivo 300, una caja adicional (esto es, la caja de tabla de muestras 309) que documenta todas las muestras 305 que contienen al menos una imagen IRAP. En otras palabras, la caja adicional identifica todas las muestras 305 que contienen al menos una imagen IRAP. En el ejemplo de la FIG. 5A, la caja adicional define un grupo de muestra que documenta (p. ej., identifica) todas las muestras 305 que contienen al menos una imagen IRAP. En otras palabras, la caja adicional especifica que las muestras 305 que contienen al menos una imagen IRAP pertenecen a un grupo de muestra.

De acuerdo con las técnicas de esta divulgación, la caja SampleGroupDescription 312 puede incluir una caja oinf 316. La caja oinf puede almacenar información de formato de representación para cada punto de operación de los datos de vídeo. La información de formato de representación puede incluir uno o más de una resolución espacial, una profundidad de bit, o un formato de color. Adicionalmente, la caja oinf puede almacenar un recuento de capas que indica un número de capas necesarias de un punto de operación de los datos de vídeo. La caja oinf puede adicionalmente almacenar información de tasa de bits para cada punto de operación de los datos de vídeo. Por lo tanto, puede no existir ninguna necesidad de señalizar una caja de tasa de bits tras una caja de configuración debido a la información de tasa de bits que es señalada en la caja oinf.

Adicionalmente, puede no existir ninguna necesidad de almacenar información de perfil, capa, y nivel PTL, información de formato de representación, e información de tasa de fotogramas en un registro de configuración de decodificador del formato de archivo. Toda la demás información en el registro de configuración de decodificador puede estar asociada con todas las capas de los datos de vídeo en una pista. Un registro de configuración de decodificador para cada capa de los datos de vídeo puede almacenar información de formato de representación e información de tasa de fotogramas. El registro de configuración de decodificador puede almacenar información de paralelismo para cada capa de los datos de vídeo. Típicamente los archivos solo incluyen un registro de configuración de decodificador para una pista, pero una pista puede contener una o más capas y uno o más puntos de operación. La información de PTL, la información de formato de representación, y la información de tasa de fotogramas puede ser asociadas con o bien cada capa o cada punto de operación (OP). Por lo tanto, a diferencia del formato de archivo HEVC que solo soporta una capa, un registro de configuración de decodificador puede no ser capaz de facilitar apropiadamente esta asociación para el formato de archivo LHEVC que soporta múltiples capas.

El registro de configuración de decodificador puede no almacenar un índice de punto de operación en un registro de configuración de decodificador, donde un índice de punto de operación se refiere a un índice del punto de operación documentado en la caja de información de punto de operación. Almacenar un índice de punto de operación en un registro de configuración de decodificador puede hacer que un dispositivo que reproduce una pista (esto es, el asociado con ese registro de configuración de decodificador) reproduzca el punto de operación referido por ese índice de punto de operación. Sin embargo, puede haber más puntos de operación disponibles. Eliminar el índice de punto

de operación puede habilitar mejor un dispositivo de reproducción para que identifique todos los puntos de operación soportados por un archivo. El registro de configuración de decodificador puede almacenar una lista de índices de punto de operación asociados con una pista de los datos de vídeo. El registro de configuración de decodificador puede, por ejemplo, ser derivado de información en la caja de entrada de muestras 311 de la FIG. 5A.

Un registro de configuración de decodificador almacena información tal como el tamaño de un campo de longitud usado en cada muestra para indicar la longitud de sus unidades NAL contenidas así como los conjuntos de parámetros, si están almacenados en la entrada de muestra. Un registro de configuración de decodificador puede, por ejemplo, estar enmarcado externamente (p. ej., su tamaño debe ser proporcionado por la estructura que lo contiene). El registro de configuración de decodificador también puede contener un campo de versión para identificar una versión de una especificación que es seguida, con cambios incompatibles con el registro que es indicado por un cambio de número de versión. En cambio, las extensiones compatibles con este registro pueden no necesitar un cambio del código de versión de configuración. El registro de configuración de decodificador también puede incluir valores para varios elementos de sintaxis HEVC tales como general_profile_space, general_tier_flag, general_profile_idc, general_profile_compatibility_flags, general_constraint_indicator_flags, general_level_idc, min_spatial_segmentation_idc, chroma_format_idc, bit_depth_luma_minus8 y bit_depth_chroma_minus8, que están definidos en HEVC. Un registro de configuración de decodificador puede contener información general que asocia, con la pista que contiene el registro de configuración, el número de subcapas temporales, información de segmentación, tipo de paralelismo soportado, y unidades NAL de conjuntos de parámetros (p. ej., VPS, SPS, PPS, SEI, etc.).

Además, de conformidad con una o más técnicas de esta divulgación, cada una de las cajas de entrada de muestras 311 puede incluir un valor (p. ej., all_pics_are_IRAP) que indica si todas las imágenes codificadas en la muestra correspondiente son imágenes IRAP. En algunos ejemplos, el valor que es igual a 1 especifica que no todas las imágenes codificadas en la muestra son imágenes IRAP. El valor que es igual a 0 especifica que no se requiere que cada imagen codificada en cada muestra del grupo de muestra sea una imagen IRAP.

En algunos ejemplos, cuando no todas las imágenes codificadas en una muestra particular son imágenes IRAP, el dispositivo de generación de archivos 34 puede incluir, en una de las cajas de entrada de muestras 311 para la muestra particular, un valor (p. ej., num_IRAP_pics) que indica un número de imágenes IRAP en la muestra particular. Adicionalmente, el dispositivo de generación de archivos 34 puede incluir, en la entrada de muestra para la muestra particular, valores que indican identificadores de capa de imágenes IRAP en la muestra particular. El dispositivo de generación de archivos 34 también puede incluir, en la entrada de muestra para la muestra particular, un valor que indica un tipo de unidad NAL de unidades NAL VCL en imágenes IRAP de la muestra particular.

Además, en el ejemplo de la FIG. 5A, la caja de tabla de muestras 309 incluye una caja de información de submuestra 314. Aunque el ejemplo de la FIG. 5A solo muestra una caja de información de submuestra, la caja de tabla de muestras 309 puede incluir múltiples cajas de información de submuestra. En general, una caja de información de submuestra está diseñada para contener información de submuestra. Una submuestra es un rango contiguo de bytes de una muestra. ISO/IEC 14496-12 indica que la definición específica de una submuestra será proporcionada para un sistema de codificación dado, tal como H.264/AVC o HEVC.

La sección 8.4.8 de ISO/IEC 14496-15 especifica una definición de una submuestra para HEVC. Particularmente, la sección 8.4.8 de ISO/IEC 14496-15 especifica que para el uso de la caja de información de submuestra (8.7.7 de ISO/IEC 14496-12) en un flujo HEVC, una submuestra está definida sobre la base de un valor de un campo de banderas de la caja de información de submuestra. De conformidad con una o más técnicas de esta divulgación, si el campo de banderas en la caja de información de submuestra 314 es igual a 5, una submuestra que corresponde a la caja de información de submuestra 314 contiene una imagen codificada y las unidades NAL no VCL asociadas. Las unidades NAL no VCL asociadas pueden incluir unidades NAL que contienen mensajes SEI aplicables a la imagen codificada y unidades NAL que contienen conjuntos de parámetros (p. ej., VPS, SPS, PPS, etc.) aplicables a la imagen codificada.

Por lo tanto, en un ejemplo, el dispositivo de generación de archivos 34 puede generar un archivo (p. ej., archivo 300) que comprende una caja de pista (p. ej., la caja de pista 306) que contiene metadatos para una pista en el archivo. En este ejemplo, los datos de medios para la pista comprenden una secuencia de muestras, siendo cada una de las muestras una unidad de acceso de vídeo de datos de vídeo multicapa (p. ej., datos de vídeo SHVC, MV-HEVC, o 3D-HEVC). Además, en este ejemplo, como parte del dispositivo de generación de archivos 34 que genera el archivo, el dispositivo de generación de archivos 34 puede generar, en el archivo, una caja de información de submuestra (p. ej., la caja de información de submuestra 314) que contiene banderas que especifican un tipo de información de submuestra dado en la caja de información de submuestra. Cuando las banderas tienen un valor particular, una submuestra que corresponde a la caja de información de submuestra contiene exactamente una imagen codificada y cero o más unidades NAL no VCL asociadas con la imagen codificada.

Además, de conformidad con una o más técnicas de esta divulgación, si el campo de banderas de la caja de información de submuestra 314 es igual a 0, la caja de información de submuestra 314 incluye además un valor DiscardableFlag, un valor NoInterLayerPredFlag, un valor LayerId, y un valor TempId. Si el campo de banderas de la

caja de información de submuestra 314 es igual a 5, la caja de información de submuestra 314 puede incluir un valor DiscardableFlag, un valor VclNalUnitType, un valor LayerId, un valor TempId, un valor NoInterLayerPredFlag, un valor SubLayerRefNalUnitFlag, y un valor reservado.

SubLayerRefNalUnitFlag igual a 0 indica que todas las unidades NAL en la submuestra son unidades NAL VCL de una imagen no de referencia de subcapa como está especificado en ISO/IEC 23008-2 (esto es, HEVC). SubLayerRefNalUnitFlag igual a 1 indica que todas las unidades NAL en la submuestra son unidades NAL VCL de una imagen de referencia de subcapa como está especificado en ISO/IEC 23008-2 (esto es, HEVC). Por lo tanto, cuando el dispositivo de generación de archivos 34 genera la caja de información de submuestra 314 y las banderas tienen un valor particular (p. ej., 5), el dispositivo de generación de archivos 34 incluye, en la caja de información de submuestra 314, una bandera adicional que indica si todas las unidades NAL en la submuestra son unidades NAL VCL de una imagen no de referencia de subcapa.

El valor DiscardableFlag indica un valor de un valor discardable_flag de las unidades NAL VCL en la submuestra. Como está especificado en la sección A.4 de ISO/IEC 14496-15, el valor discardable_flag será establecido a 1 siempre y solo si todas las unidades NAL extraídas o agregadas tienen el valor discardable_flag establecido a 1, y establecido a 0 en caso contrario. Una unidad NAL puede tener un valor discardable_flag establecido a 1 si un flujo de bits que contiene la unidad NAL puede ser correctamente decodificado sin la unidad NAL. Por lo tanto, una unidad NAL puede ser “descartable” si un flujo de bits que contiene la unidad NAL puede ser correctamente decodificado sin la unidad NAL. Todas las unidades NAL VCL en la submuestra tendrán el mismo valor discardable_flag. Por lo tanto, cuando el dispositivo de generación de archivos 34 genera la caja de información de submuestra 314 y las banderas tienen un valor particular (p. ej., 5), el dispositivo de generación de archivos 34 incluye, en la caja de información de submuestra 314, una bandera adicional (p. ej., discardable_flag) que indica si todas las unidades NAL VCL de la submuestra son descartables.

El valor NoInterLayerPredFlag indica el valor de la inter_layer_pred_enabled_flag de las unidades NAL VCL en la submuestra. La inter_layer_pred_enabled_flag será establecida a 1 siempre y solo si todas las unidades NAL VCL extraídas o agregadas tienen la inter_layer_pred_enabled_flag establecida a 1, y establecida a 0 en caso contrario. Todas las unidades NAL VCL en la submuestra tendrán el mismo valor de inter_layer_pred_enabled_flag. Por lo tanto, cuando el dispositivo de generación de archivos 34 genera la caja de información de submuestra 314 y las banderas tienen un valor particular (p. ej., 5), el dispositivo de generación de archivos 34 incluye, en la caja de información de submuestra 314, un valor adicional (p. ej., inter_layer_pred_enabled_flag) que indica si la predicción entre capas está habilitada para todas las unidades NAL VCL de la submuestra.

LayerId indica el valor nuh_layer_id de las unidades NAL en la submuestra. Todas las unidades NAL en la submuestra tendrán el mismo valor nuh_layer_id. Por lo tanto, cuando el dispositivo de generación de archivos 34 genera la caja de información de submuestra 314 y las banderas tienen un valor particular (p. ej., 5), el dispositivo de generación de archivos 34 incluye, en la caja de información de submuestra 314, un valor adicional (p. ej., LayerId) que indica un identificador de capa de cada unidad NAL de la submuestra.

TempId indica el valor TemporalId de las unidades NAL en la submuestra. Todas las unidades NAL en la submuestra tendrán el mismo valor TemporalId. Por lo tanto, cuando el dispositivo de generación de archivos 34 genera la caja de información de submuestra 314 y las banderas tienen un valor particular (p. ej., 5), el dispositivo de generación de archivos 34 incluye en, la capa de información de submuestra 314, un valor adicional (p. ej., TempId) que indica un identificador temporal de cada unidad NAL de la submuestra.

VclNalUnitType indica el elemento de sintaxis nal_unit_type de las unidades NAL VCL en la submuestra. El elemento de sintaxis nal_unit_type es un elemento de sintaxis en una cabecera de unidad NAL de una unidad NAL. El elemento de sintaxis nal_unit_type especifica el tipo de la RBSP contenida en la unidad NAL. Todos los nal_unit_type de las unidades NAL VCL en la submuestra tendrán el mismo valor nal_unit_type. Por lo tanto, cuando el dispositivo de generación de archivos 34 genera la caja de información de submuestra 314 y las banderas tienen un valor particular (p. ej., 5), el dispositivo de generación de archivos 34 incluye, en la caja de información de submuestra 314, un valor adicional (p. ej., VclNalUnitType) que indica un tipo de unidad NAL de las unidades NAL VCL de la submuestra. Todas las unidades NAL VCL de la submuestra tienen el mismo tipo de unidad NAL.

La FIG. 5B es un diagrama conceptual que ilustra una estructura de ejemplo alternativa del archivo 300, de conformidad con una o más técnicas de esta divulgación. En el ejemplo de la FIG. 5B, en vez de estar la caja oinf 316 incluida en la caja de descripción de grupo de muestra 312, como está mostrado en la FIG. 5A, la caja oinf 316 está incluida en la caja de información de medios 308 como una caja separada de la caja de tabla de muestras 309. El contenido y la función de las varias cajas en la FIG. 3B pueden de otro modo ser el mismo como fue descrito con respecto a la FIG. 5A.

La FIG. 6 es un diagrama conceptual que ilustra una estructura de ejemplo de un archivo 300, de conformidad con una o más técnicas de esta divulgación. Como está especificado en la sección 8.4.9 de ISO/IEC 14496-15, HEVC permite muestras de formato de archivo que son usadas solo para referencia y no salida. Por ejemplo, HEVC permite una imagen de referencia no mostrada en vídeo.

Además, la sección 8.4.9 de ISO/IEC 14496-15 especifica que cuando cualquiera de tal muestra de no salida está presente en una pista, el archivo estará restringido como sigue:

1. A una muestra de no salida se le dará un tiempo de composición que está fuera del rango de tiempo de las muestras que son transmitidas.
2. Una lista de edición será usada para excluir los tiempos de composición de las muestras de no salida.
3. Cuando la pista incluye una CompositionOffsetBox (‘ctts’),

a. será usada la versión 1 de la CompositionOffsetBox,
b. el valor de sample_offset será establecido igual a -2^{31} para cada muestra de no salida,
c. la CompositionToDecodeBox (‘cslg’) debe ser contenida en la SampleTableBox (‘stbl’) de la pista, y
d. cuando la CompositionToDecodeBox está presente para la pista, el valor del campo leastDecodeToDisplayDelta en la caja será igual al desplazamiento de composición más pequeño en la CompositionOffsetBox excluyendo los valores sample_offset para muestras de no salida.

NOTA: por lo tanto, leastDecodeToDisplayDelta es mayor que -2^{31}.

Como está especificado en ISO/IEC 14496-12, la CompositionOffsetBox proporciona el desplazamiento entre el tiempo de decodificación y el tiempo de composición. La CompositionOffsetBox incluye un conjunto de valores sample_offset. Cada uno de los valores sample_offset es un entero no negativo que da el desplazamiento entre el tiempo de composición y el tiempo de decodificación. El tiempo de composición se refiere a un tiempo en el cual una muestra debe ser transmitida. El tiempo de decodificación se refiere a un tiempo en el cual una muestra debe ser decodificada.

Como está indicado arriba, una unidad NAL de segmento codificado puede incluir una cabecera de segmento de segmento. La cabecera de segmento de segmento puede ser parte de un segmento de segmento codificado y puede contener elementos de datos que pertenecen a la primera o todas las CTU en el segmento de segmento. En HEVC, la cabecera de segmento de segmento incluye un elemento de sintaxis pic_output_flag. En general, el elemento de sintaxis pic_output_flag está incluido en una primera cabecera de segmento de segmento de un segmento de una imagen. Por consiguiente, esta divulgación se puede referir a la pic_output_flag de la primera cabecera de segmento de segmento del segmento de la imagen como la pic_output_flag de la imagen.

Como está especificado en la sección 7.4.7.1 del HEVC WD, el elemento de sintaxis pic_output_flag afecta la transmisión de imagen codificada y los procesos de eliminación como están especificados en el Anexo C de HEVC WD. En general, si el elemento de sintaxis pic_output_flag de una cabecera de segmento de segmento para un segmento de segmento es 1, una imagen que incluye un segmento que corresponde a la cabecera de segmento de segmento es transmitida. En caso contrario, si el elemento de sintaxis pic_output_flag de la cabecera de segmento de segmento para un segmento de segmento es 0, la imagen que incluye el segmento que corresponde a la cabecera de segmento de segmento puede ser codificada para el uso como una imagen de referencia, pero no es transmitida.

De conformidad con una o más técnicas de esta divulgación, las referencias en la sección 8.4.9 de ISO/IEC 14496-15 a HEVC pueden ser reemplazadas con referencias correspondientes a SHVC, MV-HEVC, o 3D-HEVC. Además, de conformidad con una o más técnicas de esta divulgación, cuando una unidad de acceso contiene algunas imágenes codificadas que tienen pic_output_flag igual a 1 y algunas otras imágenes codificadas que tienen pic_output_flag igual a 0, al menos dos pistas deben ser usadas para almacenar el flujo. Para respectivamente cada una de las pistas, todas las imágenes codificadas en cada muestra de la pista respectiva tienen el mismo valor de pic_utput_flag. Por lo tanto, todas las imágenes codificadas en una primera de las pistas tienen pic_output_flag igual a 0 y todas las imágenes codificadas en una segunda de las pistas tienen pic_output_flag igual a 1.

Correspondientemente, en el ejemplo de la FIG. 6, el dispositivo de generación de archivos 34 puede generar un archivo 400. Similarmente al archivo 300 en el ejemplo de la FIG. 5A, el archivo 400 incluye una caja de película 402 y una o más cajas de datos de medios 404. Cada una de las cajas de datos de medios 404 puede corresponder a una pista diferente del archivo 400. La caja de película 402 puede contener metadatos para pistas del archivo 400. Cada pista del archivo 400 puede comprender un flujo continuo de datos de medios. Cada una de las cajas de datos de medios 404 puede incluir una o más muestras 405. Cada una de las muestras 405 puede comprender una unidad de acceso de audio o vídeo.

Como está indicado arriba, en algunos ejemplos, cuando una unidad de acceso contiene algunas imágenes codificadas que tienen pic_output_flag igual a 1 y algunas otras imágenes codificadas que tienen pic_output_flag igual a 0, al menos dos pistas deben ser usadas para almacenar el flujo. Correspondientemente, en el ejemplo de la FIG. 6, la caja de película 402 incluye una caja de pista 406 y una caja de pista 408. Cada una de las cajas de pista 406 y 408 encierran metadatos para una pista diferente del archivo 400. Por ejemplo, la caja de pista 406 puede encerrar metadatos para una pista que tiene imágenes codificadas con pic_output_flag igual a 0, y ninguna imagen con pic_output_flag igual a 1. La caja de pista 408 puede encerrar metadatos para una pista que tiene imágenes codificadas con pic_output_flag igual a 1, y ninguna imagen con pic_output_flag igual a 0.

Por lo tanto, en un ejemplo, el dispositivo de generación de archivos 34 puede generar un archivo (p. ej., el archivo 400) que comprende una caja de datos de medios (p. ej., la caja de datos de medios 404) que encierra (p. ej., comprende) contenido de medios. El contenido de medios comprende una secuencia de muestras (p. ej., las muestras 405). Cada una de las muestras puede ser una unidad de acceso de datos de vídeo multicapa. En este ejemplo, cuando el dispositivo de generación de archivos 34 genera el archivo, en respuesta a una determinación de que al menos una unidad de acceso del flujo de bits incluye una imagen codificada que tiene una bandera de transmisión de imagen igual a 1 y una imagen codificada que tiene una bandera de transmisión de imagen igual a 0, el dispositivo de generación de archivos 34 puede usar al menos dos pistas para almacenar el flujo de bits en el archivo. Para cada pista respectiva de las al menos dos pistas, todas las imágenes codificadas en cada muestra de la pista respectiva tienen el mismo valor de la bandera de transmisión de imagen. Las imágenes que tienen banderas de transmisión de imagen igual a 1 se pueden transmitir y las imágenes que tienen banderas de transmisión de imagen igual a 0 se pueden usar como imágenes de referencia pero no pueden ser transmitidas.

La FIG. 7 es un diagrama de flujos que ilustra una operación de ejemplo del dispositivo de generación de archivos 34, de conformidad con una o más técnicas de esta divulgación. La operación de la FIG. 7, junto con las operaciones ilustradas en otros diagramas de flujos de esta divulgación, son ejemplos. Otras operaciones de ejemplo de conformidad con las técnicas de esta divulgación pueden incluir más, menos, o diferentes acciones.

En el ejemplo de la FIG. 7, el dispositivo de generación de archivos 34 genera un archivo. Como parte de generar el archivo, el dispositivo de generación de archivos 34 obtiene datos de vídeo multicapa (170) y almacena los datos de vídeo multicapa en un formato de archivo (172). El dispositivo de generación de archivos 34 almacena información de formato de representación para cada punto de operación de los datos de vídeo multicapa en una caja oinf del formato de archivo (174). El dispositivo de generación de archivos 34 genera un archivo de datos de vídeo formateado de acuerdo con el formato de archivo (176). La información de formato de representación puede incluir uno o más de una resolución espacial, una profundidad de bit, o un formato de color. El dispositivo de generación de archivos 34 puede adicionalmente o alternativamente almacenar información de tasa de bits para cada punto de operación de los datos de vídeo multicapa en la caja oinf del formato de archivo y/o puede no señalizar una caja de tasa de bits tras una caja de configuración del formato de archivo. El dispositivo de generación de archivos 34 puede adicionalmente o alternativamente no almacenar información de perfil, capa, y nivel (PTL), información de formato de representación, e información de tasa de fotogramas en un registro de configuración de decodificador del formato de archivo y asociar toda la otra información en el registro de configuración de decodificador con todas las capas de los datos de vídeo multicapa en una pista. El dispositivo de generación de archivos 34 puede adicionalmente o alternativamente almacenar un recuento de capa en la caja oinf del formato de archivo, en donde el recuento de capa indica un número de capas necesarias de un punto de operación de los datos de vídeo multicapa.

La caja oinf puede estar incluida en una caja de información de medios, y la caja oinf puede estar incluida en una caja de descripción de grupo de muestra. La caja de descripción de grupo de muestra puede estar incluida en una caja de tabla de muestras, y la caja de tabla de muestras puede estar incluida en la caja de información de medios.

El dispositivo de generación de archivos 34 puede almacenar información de formato de representación e información de tasa de fotogramas en un registro de configuración de decodificador para cada capa de los datos de vídeo multicapa. El dispositivo de generación de archivos 34 puede adicionalmente o alternativamente almacenar información de paralelismo en el registro de configuración de decodificador para cada capa de los datos de vídeo multicapa. El dispositivo de generación de archivos 34 puede no almacenar un índice de punto de operación en un registro de configuración de decodificador del formato de archivo. El dispositivo de generación de archivos 34 puede adicionalmente o alternativamente almacenar una lista de índices de punto de operación asociada con una pista de los datos de vídeo multicapa en un registro de configuración de decodificador del formato de archivo.

La FIG. 8 es un diagrama de flujos que ilustra una operación de ejemplo de un dispositivo de lectura de archivos, tal como el dispositivo de destino 14, la entidad de posprocesamiento 27, o la entidad de red 29. La operación de la FIG. 8, junto con las operaciones ilustradas en otros diagramas de flujos de esta divulgación, son ejemplos. Otras operaciones de ejemplo de conformidad con las técnicas de esta divulgación pueden incluir más, menos, o diferentes acciones.

En el ejemplo de la FIG. 8, un dispositivo de lectura de archivos obtiene un archivo de datos de vídeo multicapa formateado de acuerdo con un formato de archivo (180). El dispositivo de lectura de archivos, para el formato de archivo, determina información de formato de representación para cada punto de operación de los datos de vídeo multicapa en una caja oinf para el formato de archivo (182). El dispositivo de lectura de archivos, posiblemente en conjunción con un decodificador de vídeo tal como el decodificador de vídeo 30, decodifica los datos de vídeo multicapa basándose en la información de formato de representación (184) determinada.

En uno o más ejemplos, las funciones descritas pueden ser implementadas en hardware, software, firmware, o cualquier combinación de estos. Si se implementan en software, las funciones pueden ser almacenadas en o transmitidas a través de, como una o más instrucciones o código, un medio legible por ordenador y ejecutadas por una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de

almacenamiento legibles por ordenador, los cuales corresponden a un medio tangible tal como medios de almacenamiento de datos, o medios de comunicación que incluyen cualquier medio que facilita la transferencia de un programa de ordenador de un lugar a otro, p. ej., de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por ordenador pueden generalmente corresponder a (1) medios de almacenamiento legibles por ordenador tangibles que son no transitorios o (2) un medio de comunicación tal como una señal u onda portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible que puede ser accedido por 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 de ordenador puede incluir un medio legible por ordenador.

A modo de ejemplo, y no de limitación, tales medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento de disco óptico, almacenamiento de disco magnético, u otros dispositivos de almacenamiento magnético, memoria flash, o cualquier otro medio que puede ser usado para almacenar código de programa deseado en la forma de instrucciones o estructuras de datos y que pueden ser accedidos por un ordenador. También, cualquier conexión está adecuadamente designada como medio legible por ordenador. Por ejemplo, si las instrucciones son transmitidas desde un sitio web, servidor, u otra fuente remota usando un cable coaxial, cable de fibra óptica, par trenzado, línea de abonado digital (DSL), o tecnologías inalámbricas tales como infrarrojos, radio, y microondas, entonces el cable coaxial, cable de fibra óptica, par trenzado, DSL, o tecnologías inalámbricas tales como infrarrojos, radio, y microondas están incluidas en la definición de medio. Debe entenderse, sin embargo, 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 en cambio dirigidos a medios de almacenamiento tangibles no transitorios. De disco y disco, como están usados aquí, incluye disco compacto (CD), disco láser, disco óptico, disco versátil digital (DVD), disquete y disco Blu-Ray, donde las unidades de disco usualmente reproducen datos magnéticamente, mientras que los discos reproducen datos ópticamente con láseres. Las combinaciones de lo de arriba deben también ser incluidas dentro del alcance de medios legibles por ordenador.

Las instrucciones pueden ser ejecutadas por uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de uso general, circuitos integrados de aplicación específica (ASIC), matrices de puertas programables en campo (FPGA), u otra circuitería de lógica discreta o integrada equivalente. Correspondientemente, el término “procesador”, como está usado aquí puede referirse a cualquiera de las estructuras anteriores o cualquier otra estructura adecuada para la implementación de las técnicas descritas aquí. Además, en algunos aspectos, la funcionalidad descrita aquí puede estar proporcionada dentro de módulos de software y/o hardware dedicados configurados para codificar y decodificar, o incorporada en un códec combinado. También, las técnicas podrían ser plenamente implementadas en uno o más circuitos o elementos lógicos.

Las técnicas de esta divulgación pueden ser implementadas en una amplia variedad de dispositivos o aparatos, incluido un terminal inalámbrico, un circuito integrado (IC) o un conjunto de IC (p. ej., un conjunto de chips). Varios componentes, módulos, o unidades están descritos en esta divulgación para enfatizar aspectos funcionales de los dispositivos configurados para llevar a cabo las técnicas divulgadas, pero no requieren necesariamente la realización por parte de diferentes unidades de hardware. Más bien, como está descrito arriba, varias unidades pueden ser combinadas en una unidad de hardware códec o proporcionadas por una colección de unidades de hardware interoperativas, incluidos uno o más procesadores como está descrito arriba, en conjunción con software y/o firmware adecuado.

Varios ejemplos han sido descritos. Estos y otros ejemplos están dentro del alcance de las siguientes reivindicaciones.

REIVINDICACIONES

1. Un método de procesar datos de vídeo multicapa que comprende una serie de unidades de capa de abstracción de red, NAL, teniendo cada una un identificador de capa y un identificador temporal, en donde cada capa de los datos de vídeo multicapa es definida como un conjunto de unidades NAL que tienen el mismo identificador de capa, comprendiendo el método:

obtener (170) datos de vídeo multicapa que comprenden más de un punto de operación, en donde cada punto de operación tiene un conjunto de identificadores de capa y un identificador temporal, e incluye cada unidad NAL cuyo identificador de capa está incluido en el conjunto de identificadores de capa del punto de operación y cuyo identificador temporal es menor que o igual al identificador temporal del punto de operación;
almacenar (172) los datos de vídeo multicapa en un formato de archivo, en donde el formato de archivo incluye una caja de información de puntos de operación (oinf) (316) que identifica los puntos de operación incluidos en los datos de vídeo multicapa;
almacenar un indicador de información de perfil, capa, y nivel (PTL) para cada capa de cada punto de operación de los datos de vídeo multicapa en la caja oinf; y
generar un archivo de datos de vídeo formateado de acuerdo con el formato de archivo
caracterizado por que el método comprende además:
almacenar (174) información de formato de representación para cada punto de operación de los datos de vídeo multicapa en la caja oinf, en donde almacenar la información de formato de representación comprende almacenar solo un conjunto de información de formato de representación para cada punto de operación, y en donde cada conjunto de información de formato de representación comprende uno o más de una resolución espacial, una profundidad de bit, o un formato de color.

2. El método de la reivindicación 1, que comprende además almacenar información de tasa de bits para cada punto de operación de los datos de vídeo multicapa en la caja oinf (316) del formato de archivo.

3. El método de la reivindicación 1, en donde los datos de vídeo multicapa están procesados de conformidad con el formato de archivo multimedia base (ISOBMFF) de la Organización Internacional de Normalización (ISO), y en donde el formato de archivo incluye además un registro de configuración de decodificador, comprendiendo además el método:
no almacenar información de perfil, capa, y nivel (PTL), información de formato de representación, e información de tasa de fotogramas en el registro de configuración de decodificador del formato de archivo.

4. El método de la reivindicación 1, en donde los datos de vídeo multicapa están procesados de conformidad con el formato de archivo multimedia base (ISOBMFF) de la Organización Internacional de Normalización (ISO), comprendiendo además el método:
almacenar información de formato de representación e información de tasa de fotogramas en un registro de configuración de decodificador para cada capa de los datos de vídeo multicapa.

5. El método de la reivindicación 4, que comprende además:
almacenar información de paralelismo en el registro de configuración de decodificador para cada capa de los datos de vídeo multicapa.

6. El método de la reivindicación 1, en donde los datos de vídeo multicapa están procesados de conformidad con el formato de archivo multimedia base (ISOBMFF) de la Organización Internacional de Normalización (ISO), en donde el formato de archivo incluye además un registro de configuración de decodificador, comprendiendo además el método:
no almacenar un índice de punto de operación en el registro de configuración de decodificador del formato de archivo.

7. El método de la reivindicación 1, en donde los datos de vídeo multicapa están procesados de conformidad con el formato de archivo multimedia base (ISOBMFF) de la Organización Internacional de Normalización (ISO), para almacenar los datos de vídeo multicapa en un archivo (300), en donde el archivo incluye uno o más flujos de medios continuos, estando cada uno representado como una pista, comprendiendo además el método:
almacenar, en un registro de configuración de decodificador del formato de archivo, una lista de índices de punto de operación en donde los índices de punto de operación indican respectivamente los puntos de operación incluidos en una pista de los datos de vídeo multimedia.

8. El método de la reivindicación 1, que comprende además:
almacenar un recuento de capa en la caja oinf (316) del formato de archivo, en donde el recuento de capa indica el número de capas necesarias de un punto de operación de los datos de vídeo multicapa.

9. El método de la reivindicación 1, en donde la caja oinf (316) está incluida en una caja de información de medios (308).

10. El método de la reivindicación 9, en donde la caja oinf (316) está incluida en la caja de descripción de grupo de muestra (312), en donde la caja de descripción de grupo de muestra está incluida en una caja de tabla de muestras (309), y en donde la caja de tabla de muestras está incluida en la caja de información de medios (308).

11. El método de la reivindicación 1,

en donde un flujo de bits de datos de vídeo multicapa comprende un conjunto de unidades de capa de abstracción de red (NAL), teniendo cada unidad NAL un identificador de capa y un identificador temporal,
en donde el flujo de bits tiene una pluralidad de puntos de operación; y
en donde cada punto de operación corresponde a un subconjunto de las unidades NAL en el flujo de bits, incluyendo el subconjunto unidades NAL que tienen un identificador de capa en el conjunto de identificadores de capa del punto de operación y un identificador temporal menor que o igual al identificador temporal del punto de operación.

12. Un dispositivo de vídeo para procesar datos de vídeo multicapa que comprende una serie de unidades de capa de abstracción de red, NAL, teniendo cada una un identificador de capa y un identificador temporal, en donde cada capa de los datos de vídeo multicapa está definida como un conjunto de unidades NAL que tienen el mismo identificador de capa, comprendiendo el dispositivo:

medios para obtener datos de vídeo multicapa que comprenden más de un punto de operación, en donde cada punto de operación tiene un conjunto de identificadores de capa y un identificador temporal, e incluye cada unidad NAL cuyo identificador de capa está incluido en el conjunto de identificadores de capa del punto de operación y cuyo identificador temporal es menor que o igual al identificador temporal del punto de operación; y
medios para almacenar los datos de vídeo multicapa en un formato de archivo, para generar un archivo de datos de vídeo formateado de acuerdo con el formato de archivo, en donde el archivo generado incluye una caja de información de puntos de operación (oinf) que identifica los puntos de operación incluidos en los datos de vídeo multicapa, y en donde la caja oinf incluye un indicador de información de perfil, capa, y nivel (PTL) almacenada para cada capa de cada punto de operación de los datos de vídeo multicapa en la caja oinf; y
caracterizado por que el archivo generado incluye además:
información de formato de representación almacenada para cada punto de operación de los datos de vídeo multicapa en la caja oinf, en donde la información de formato de representación comprende solo un conjunto de información de formato de representación para cada punto de operación, y en donde cada conjunto de información de formato de representación comprende uno o más de una resolución espacial, una profundidad de bit, o un formato de color.

13. El dispositivo de la reivindicación 12, en donde la caja oinf está incluida en una caja de información de medios (308).

14. El dispositivo de la reivindicación 13, en donde la caja oinf está incluida además en la caja de descripción de grupo de muestra (312), en donde la caja de descripción de grupo de muestra está incluida en una caja de tabla de muestras (309), y en donde la caja de tabla de muestras está incluida en la caja de información de medios (308).

15. Un medio de almacenamiento legible por ordenador que almacena instrucciones que cuando son ejecutadas hacen que uno o más procesadores lleven a cabo el método de cualquiera de las reivindicaciones 1-11.

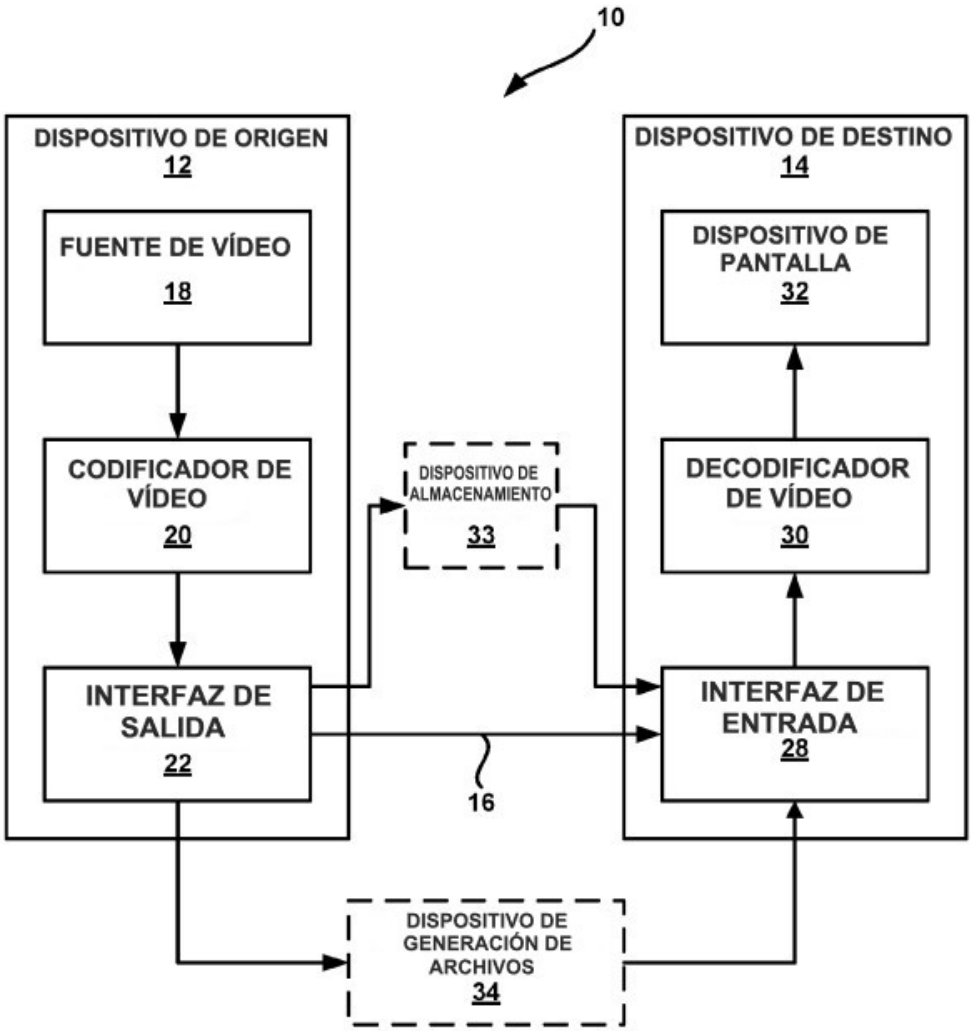
10
DISPOSITIVO DE ORIGEN
12
FUENTE DE VÍDEO
18
CODIFICADOR DE VÍDEO
20
INTERFAZ DE SALIDA
22
DISPOSITIVO DE ALMACENAMIENTO
33
16
DISPOSITIVO DE GENERACIÓN DE ARCHIVOS
34
DISPOSITIVO DE DESTINO
14
DISPOSITIVO DE PANTALLA
32
DECODIFICADOR DE VÍDEO
30
INTERFAZ DE ENTRADA
28

FIG. 1

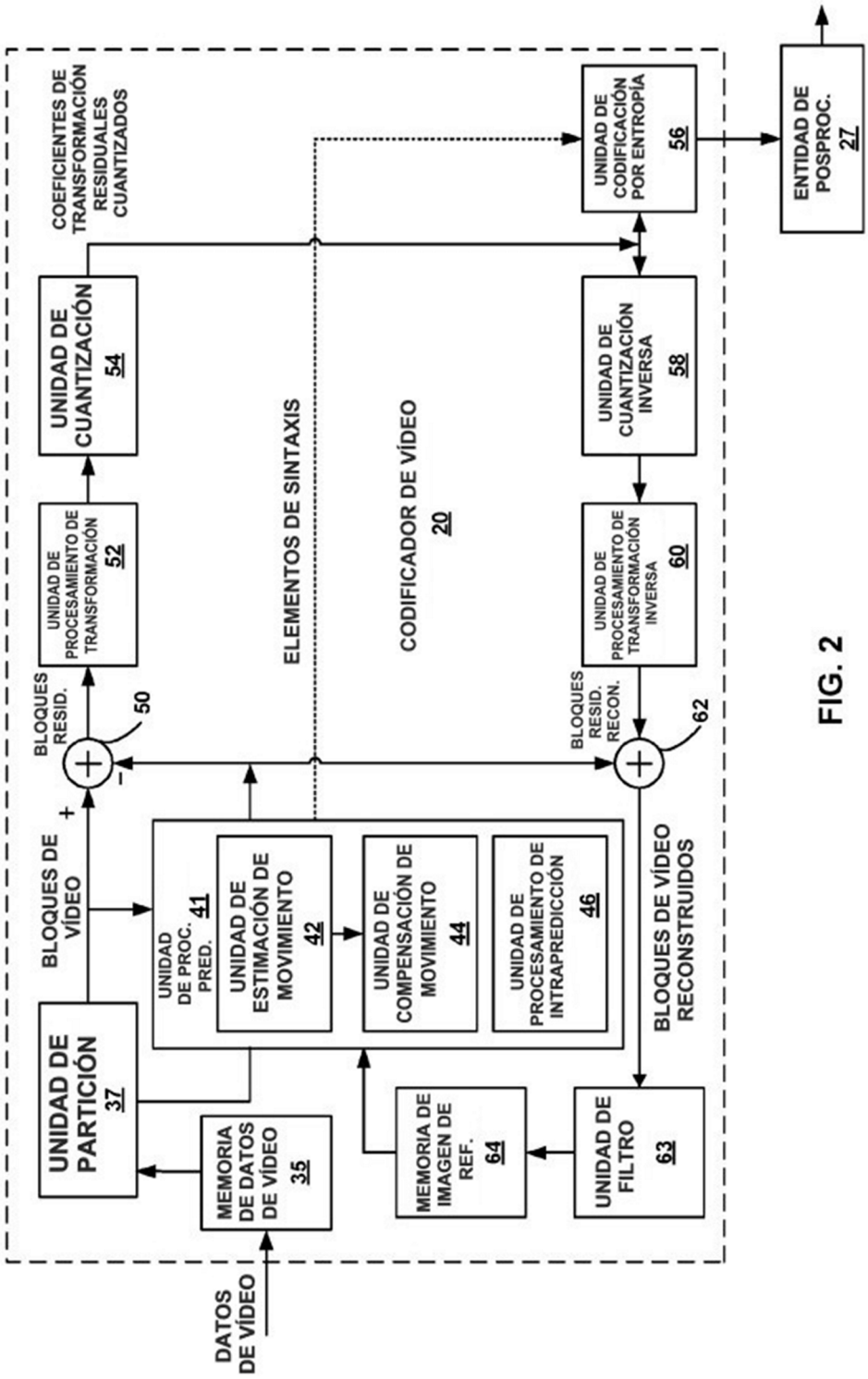
DATOS DE VÍDEO
MEMORIA DE DATOS DE VÍDEO 35
UNIDAD DE PARTICIÓN 37
BLOQUES DE VÍDEO
UNIDAD DE PROC. PRED. 41
UNIDAD DE ESTIMACIÓN DE MOVIMIENTO 42
UNIDAD DE COMPENSACIÓN DE MOVIMIENTO 44
UNIDAD DE PROCESAMIENTO DE INTRAPREDICCIÓN 46
BLOQUES RESID.
50
UNIDAD DE PROCESAMIENTO DE TRANSFORMACIÓN 52
UNIDAD DE CUANTIZACIÓN 54
COEFICIENTES DE TRANSFORMACIÓN RESIDUALES CUANTIZADOS
ELEMENTOS DE SINTAXIS
CODIFICADOR DE VÍDEO 20
UNIDAD DE CODIFICACIÓN POR ENTROPÍA 56
ENTIDAD DE POSPROC. 27
UNIDAD DE CUANTIZACIÓN INVERSA 58
UNIDAD DE PROCESAMIENTO DE TRANSFORMACIÓN INVERSA 60
BLOQUES RESID. RECON.
62
BLOQUES DE VÍDEO RECONSTRUIDOS
MEMORIA DE IMAGEN DE REF. 64
UNIDAD DE FILTRO 63

FIG. 2

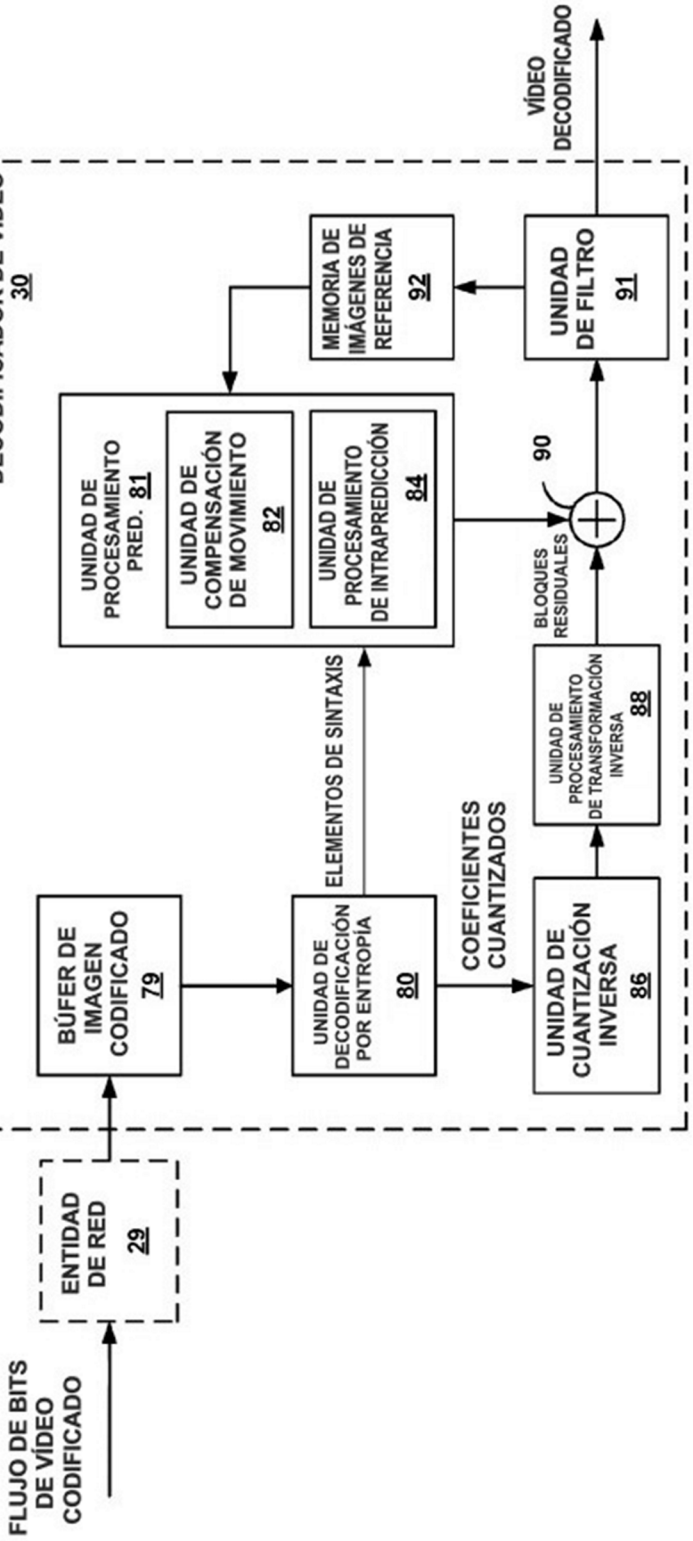
FLUJO DE BITS DE VÍDEO CODIFICADO
ENTIDAD DE RED 29
DECODIFICADOR DE VÍDEO 30
BÚFER DE IMAGEN CODIFICADO 79
UNIDAD DE DECODIFICACIÓN POR ENTROPÍA 80
ELEMENTOS DE SINTAXIS
UNIDAD DE PROCESAMIENTO PRED. 81
UNIDAD DE COMPENSACIÓN DE MOVIMIENTO 82
UNIDAD DE PROCESAMIENTO DE INTRAPREDICCIÓN 84
MEMORIA DE IMÁGENES DE REFERENCIA 92
UNIDAD DE CUANTIZACIÓN INVERSA 86
COEFICIENTES CUANTIZADOS
UNIDAD DE PROCESAMIENTO DE TRANSFORMACIÓN INVERSA 88
BLOQUES RESIDUALES
90
UNIDAD DE FILTRO 91
VÍDEO DECODIFICADO

FIG. 3

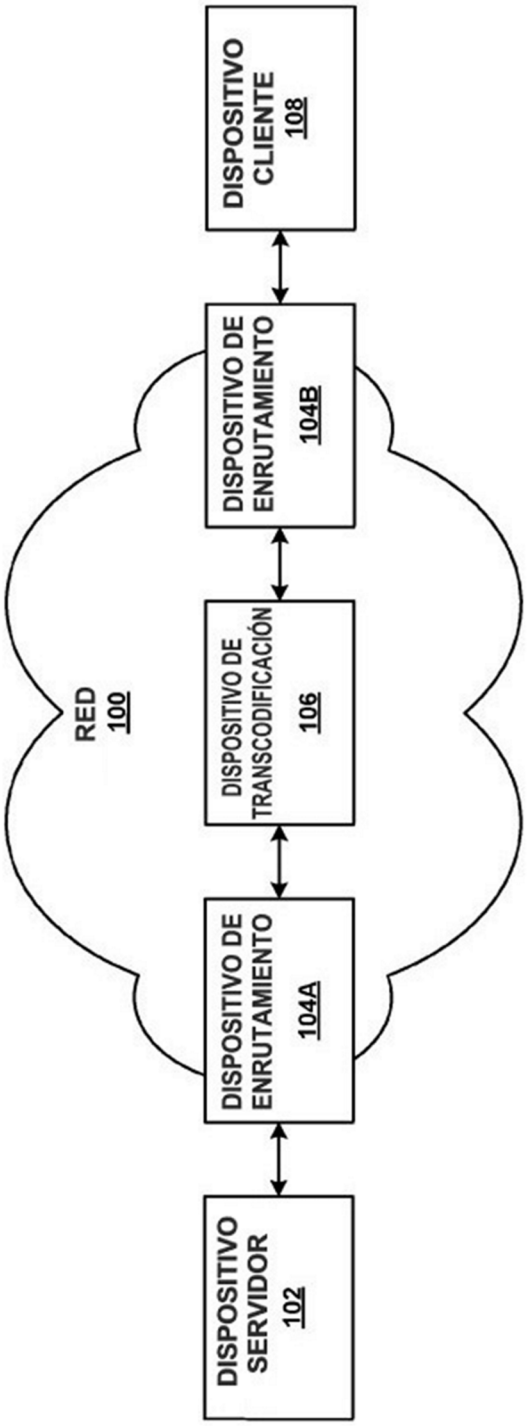
DISPOSITIVO SERVIDOR 102
DISPOSITIVO DE ENRUTAMIENTO 104A
DISPOSITIVO DE TRANSCODIFICACIÓN 106
DISPOSITIVO DE ENRUTAMIENTO 104B
DISPOSITIVO CLIENTE 108
RED 100

FIG. 4

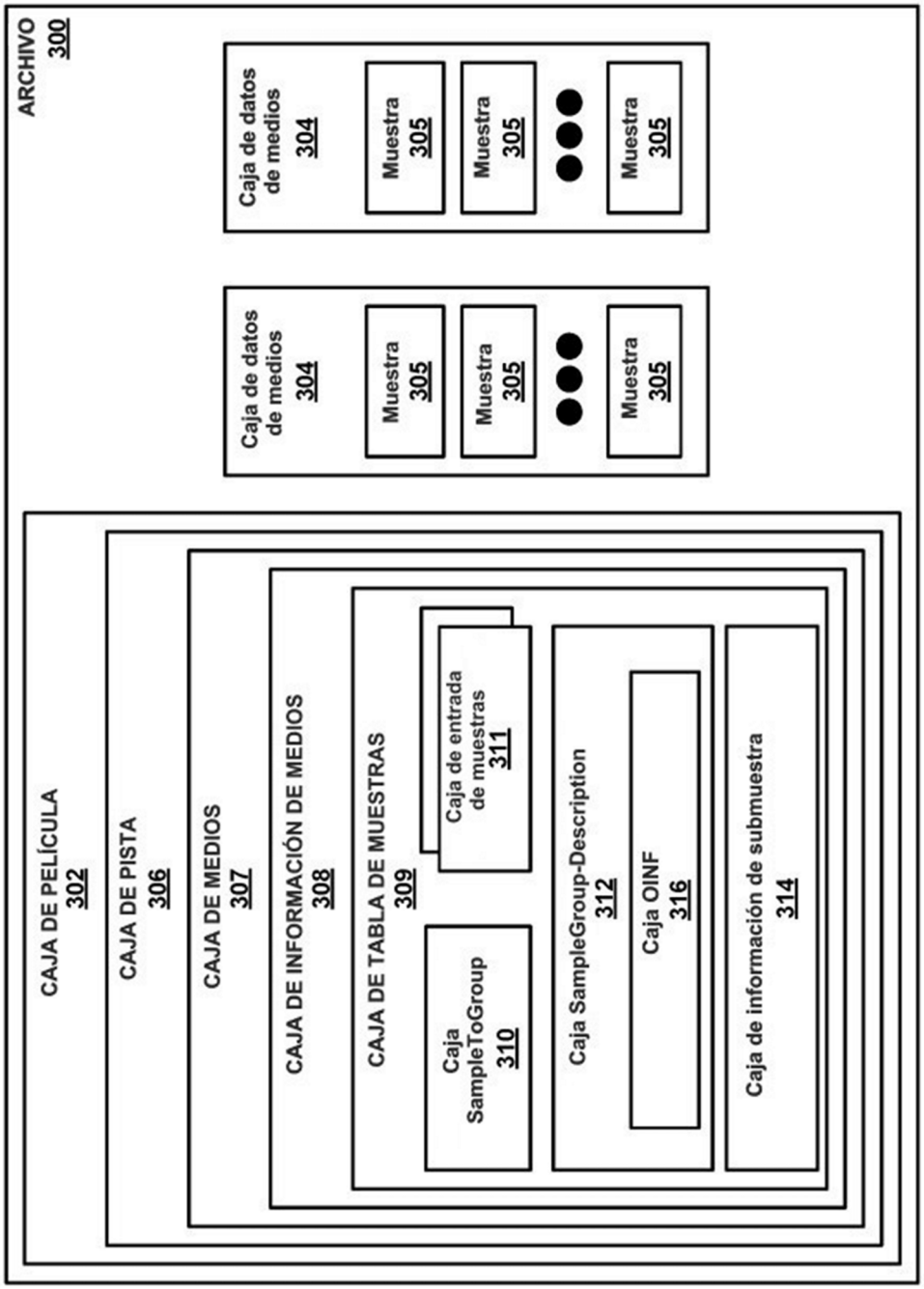
ARCHIVO
300
CAJA DE PELÍCULA
302
CAJA DE PISTA
306
CAJA DE MEDIOS
307
CAJA DE INFORMACIÓN DE MEDIOS
308
CAJA DE TABLA DE MUESTRAS
309
Caja SampleToGroup
310
Caja de entrada de muestras
311
Caja SampleGroup-Description
312
Caja OINF
316
Caja de información de submuestra
314
Caja de datos de medios
304
Muestra
305
Muestra
305
Muestra
305
Caja de datos de medios
304
Muestra
305
Muestra
305
Muestra
305

FIG. 5A

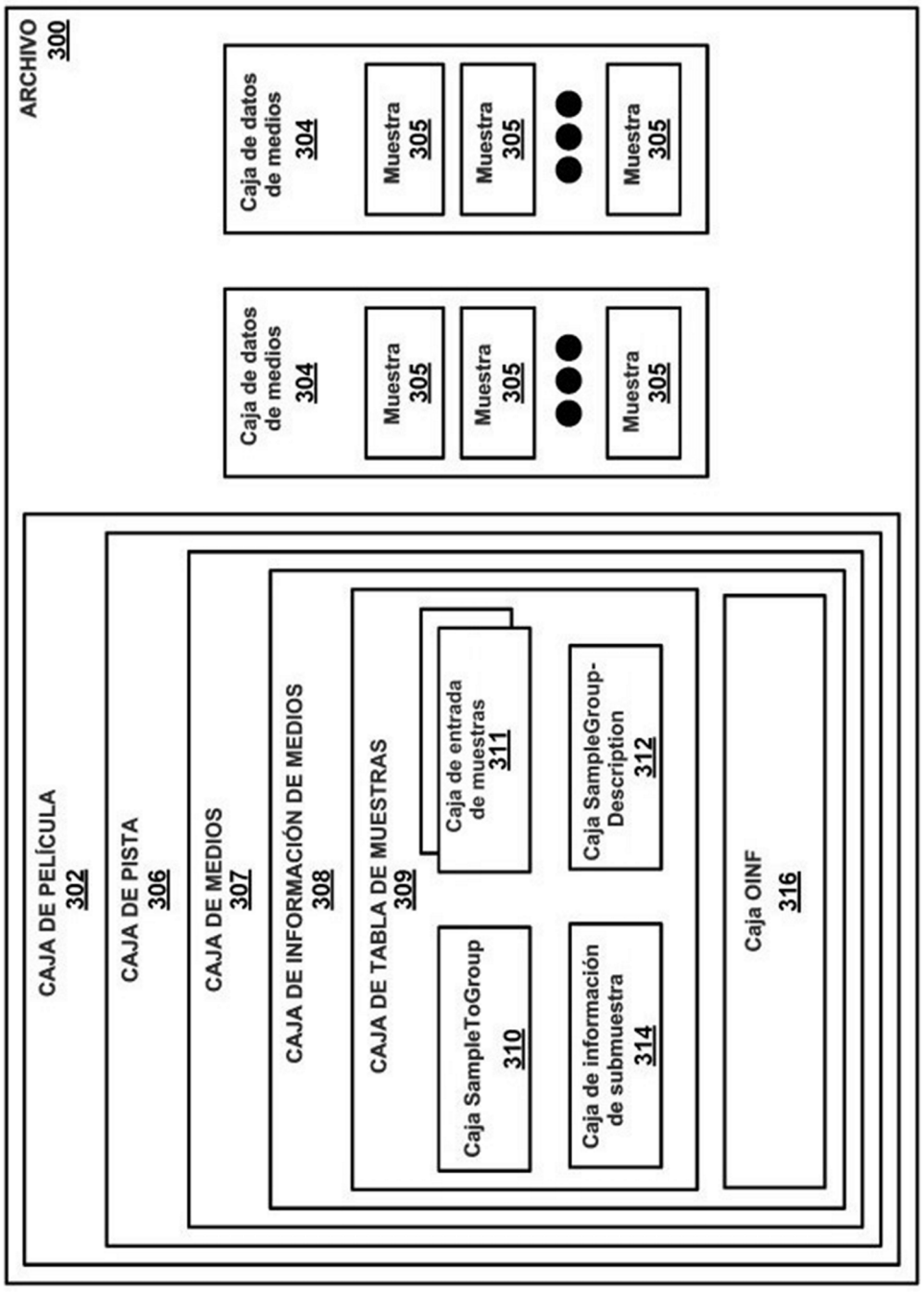
ARCHIVO
300
CAJA DE PELÍCULA
302
CAJA DE PISTA
306
CAJA DE MEDIOS
307
CAJA DE INFORMACIÓN DE MEDIOS
308
CAJA DE TABLA DE MUESTRAS
309
Caja SampleToGroup
310
Caja de entrada de muestras
311
Caja de información de submuestra
314
Caja SampleGroup-Description
312
Caja OINF
316
Caja de datos de medios
304
Muestra
305
Muestra
305
Muestra
305
Caja de datos de medios
304
Muestra
305
Muestra
305
Muestra
305

FIG. 5B

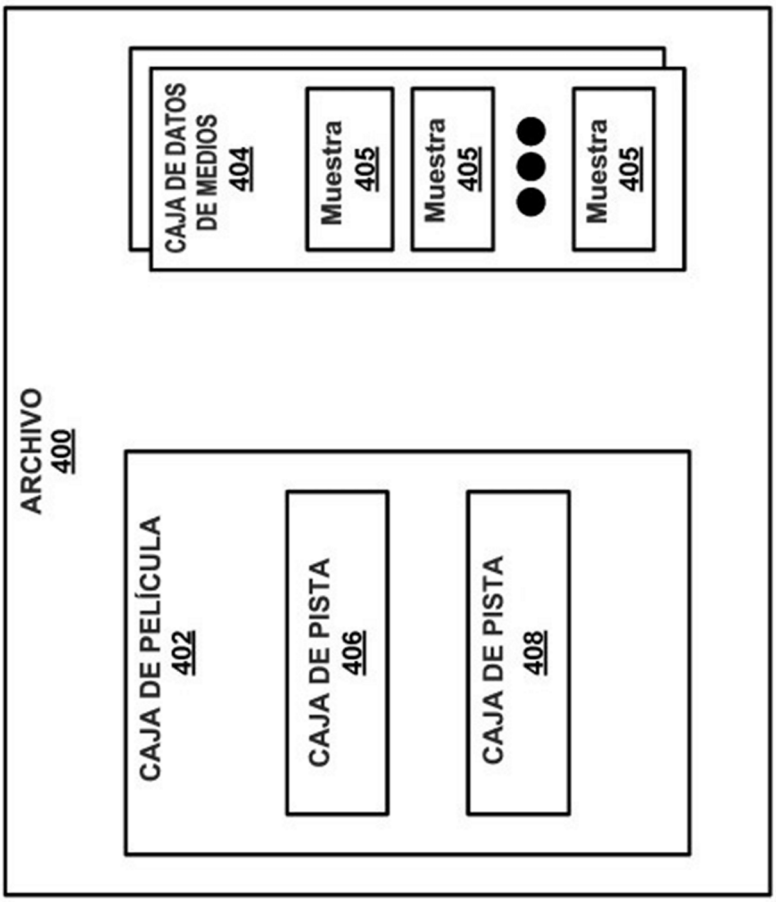
ARCHIVO
400
CAJA DE PELÍCULA
402
CAJA DE PISTA
406
CAJA DE PISTA
408
CAJA DE DATOS
DE MEDIOS
404
Muestra
405
Muestra
405
Muestra
405

FIG. 6

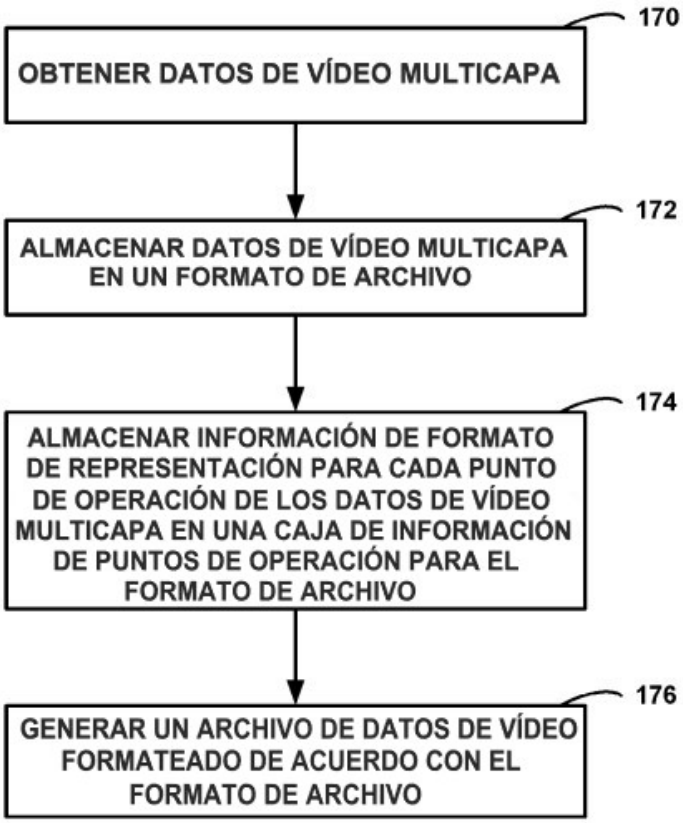
170
OBTENER DATOS DE VÍDEO MULTICAPA
172
ALMACENAR DATOS DE VÍDEO MULTICAPA EN UN FORMATO DE ARCHIVO
174
ALMACENAR INFORMACIÓN DE FORMATO DE REPRESENTACIÓN PARA CADA PUNTO DE OPERACIÓN DE LOS DATOS DE VÍDEO MULTICAPA EN UNA CAJA DE INFORMACIÓN DE PUNTOS DE OPERACIÓN PARA EL FORMATO DE ARCHIVO
176
GENERAR UN ARCHIVO DE DATOS DE VÍDEO FORMATEADO DE ACUERDO CON EL FORMATO DE ARCHIVO

FIG. 7

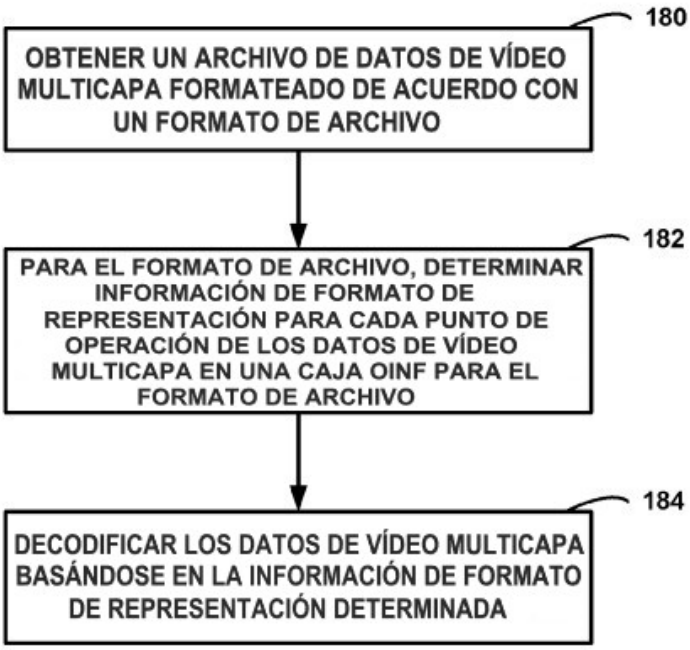
180
OBTENER UN ARCHIVO DE DATOS DE VÍDEO MULTICAPA FORMATEADO DE ACUERDO CON UN FORMATO DE ARCHIVO
182
PARA EL FORMATO DE ARCHIVO, DETERMINAR INFORMACIÓN DE FORMATO DE REPRESENTACIÓN PARA CADA PUNTO DE OPERACIÓN DE LOS DATOS DE VÍDEO MULTICAPA EN UNA CAJA OINF PARA EL FORMATO DE ARCHIVO
184
DECODIFICAR LOS DATOS DE VÍDEO MULTICAPA BASÁNDOSE EN LA INFORMACIÓN DE FORMATO DE REPRESENTACIÓN DETERMINADA

FIG. 8