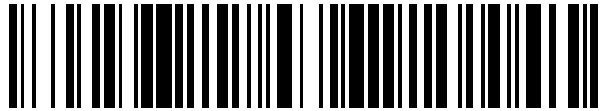


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 780 688**

51 Int. Cl.:

H04N 19/70	(2014.01)
H04N 19/463	(2014.01)
H04N 19/31	(2014.01)
H04N 19/423	(2014.01)
H04N 19/597	(2014.01)
H04N 19/44	(2014.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 86 Fecha de presentación y número de la solicitud internacional: **09.10.2014 PCT/US2014/059925**
- 87 Fecha y número de publicación internacional: **16.04.2015 WO15054509**
- 96 Fecha de presentación y número de la solicitud europea: **09.10.2014 E 14789468 (7)**
- 97 Fecha y número de publicación de la concesión europea: **25.12.2019 EP 3056003**

54 Título: **Señalización para operaciones DPB basadas en un búfer de imágenes subdescodificadas (sub-DPB) en la codificación de vídeo**

30 Prioridad:

10.10.2013 US 201361889515 P
14.10.2013 US 201361890841 P
08.10.2014 US 201414509797

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
26.08.2020

73 Titular/es:

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

72 Inventor/es:

RAMASUBRAMONIAN, ADARSH KRISHNAN y
WANG, YE-KUI

74 Agente/Representante:

FORTEA LAGUNA, Juan José

ES 2 780 688 T3

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

DESCRIPCIÓN

Señalización para operaciones DPB basadas en un búfer de imágenes subdescodificadas (sub-DPB) en la codificación de vídeo

5

CAMPO TÉCNICO

[0001] Esta divulgación se refiere a la codificación de vídeo.

10 ANTECEDENTES

[0002] Las capacidades del vídeo digital pueden incorporarse en una amplia gama de dispositivos, incluyendo televisores digitales, sistemas de difusión directa digital, sistemas de difusión inalámbrica, asistentes digitales personales (PDA), ordenadores portátiles o de escritorio, tabletas electrónicas, lectores de libros electrónicos, cámaras digitales, dispositivos de grabación digital, reproductores de medios digitales, dispositivos de videojuegos, consolas de videojuegos, teléfonos celulares o de radio por satélite, los denominados "teléfonos inteligentes", dispositivos de videoconferencia, dispositivos de transmisión continua de vídeo y similares. Los dispositivos de vídeo digital implementan técnicas de codificación de vídeo, tales como las descritas en las normas definidas por MPEG-2, MPEG-4, ITU-T H.263, ITU-T H.264/MPEG-4, Parte 10, Codificación Avanzada de Vídeo (AVC), la norma de Codificación de Vídeo de Alta Eficacia (HEVC), actualmente en desarrollo, y las ampliaciones de tales normas. Los dispositivos de vídeo pueden transmitir, recibir, codificar, descodificar y/o almacenar información de vídeo digital más eficazmente, implementando dichas técnicas de codificación de vídeo.

[0003] Las técnicas de codificación de vídeo incluyen la predicción espacial (intraimagen) y/o la predicción temporal (interimagen) para reducir o eliminar la redundancia intrínseca en las secuencias de vídeo. Para la codificación de vídeo basada en bloques, un fragmento de vídeo (por ejemplo, una trama de vídeo o una parte de una trama de vídeo) puede dividirse en bloques de vídeo, que también pueden denominarse bloques de árbol, unidades de codificación (CU) y/o nodos de codificación. Los bloques de vídeo en un fragmento intracodificado (I) de una imagen se codifican usando predicción espacial con respecto a muestras de referencia en bloques vecinos en la misma imagen. Los bloques de vídeo de un fragmento intercodificado (P o B) de una imagen pueden usar predicción espacial con respecto a unas muestras de referencia de bloques vecinos de la misma imagen, o predicción temporal con respecto a unas muestras de referencia de otras imágenes de referencia. Las imágenes pueden denominarse tramas y las imágenes de referencia pueden denominarse tramas de referencia.

[0004] La predicción espacial o temporal da como resultado un bloque predictivo para un bloque a codificar. Los datos residuales representan diferencias de píxeles entre el bloque original a codificar y el bloque predictivo. Un bloque intercodificado se codifica de acuerdo con un vector de movimiento que apunta a un bloque de muestras de referencia que forman el bloque predictivo, y los datos residuales que indican la diferencia entre el bloque codificado y el bloque predictivo. Un bloque intracodificado se codifica de acuerdo con un modo de intracodificación y de acuerdo con datos residuales. Para una mayor compresión, los datos residuales pueden transformarse desde el dominio del píxel a un dominio de transformada, dando como resultado coeficientes de transformada residuales, que a continuación se pueden cuantificar. Los coeficientes de transformada cuantificados, dispuestos inicialmente en una formación bidimensional, pueden explorarse con el fin de generar un vector unidimensional de coeficientes de transformada, y puede aplicarse codificación de entropía para lograr aún más compresión.

[0005] En "AHG9: On Signaling DPB Parameters and DPB Operation" de Deshpande (Equipo de colaboración conjunta sobre codificación de vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, 13ª reunión: Incheon, KR, 18-26 de abril de 2013, documento n.º JCTVC M0210), se propone señalar ciertos parámetros de subcapas del CVS para puntos de operación en la extensión VPS para SHVC y MV-HEVC, o en la extensión VPS para los conjuntos de capas de salida en lugar de para los puntos de operación. También se ha propuesto el proceso de operación y expulsión para el orden de salida DPB de SHVC y MV-HEVC para usar los parámetros anteriores de la extensión VPS si se señalizan para el punto de operación seleccionado (o la capa de salida establecida en la propuesta de variante). En caso contrario, se utilizan los parámetros de nivel SPS.

[0006] En "AHG9: On DPB operations in HEVC 3DV and scalable extensions" de Chen y col. (Equipo de colaboración conjunta sobre codificación de vídeo (JCT-VC) de ITU-T SG16 WP3 e ISO/IEC JTC1/SC29/WG11, 13ª reunión: Incheon, KR, 18-26 de abril de 2013, documento n.º JCTVC M0270) hay un análisis sobre las operaciones de gestión de DPB en HEVC 3DV y las extensiones ajustables a escala, que incluyen almacenamiento, marcado, salida y eliminación de imágenes descodificadas. Se informa que, con la suposición de descodificación de múltiples bucles en SHVC y la necesidad de soporte de ajustabilidad a escala espacial, la gestión de DPB que implica imágenes descodificadas de diferentes tamaños se hace necesaria. Se informa además que, en ese momento, el mecanismo para la gestión de DPB y la definición del tamaño de DPB en SHVC para la ajustabilidad a escala espacial no se especifican claramente. Se informa además que son posibles dos enfoques diferentes. Uno es a nivel de AU (con la suposición de que AU se define para incluir todas las imágenes que pertenecen a un tiempo de salida particular), el otro es a nivel de imagen. Se informa que la gestión de DPB a nivel de AU es sencilla desde el punto de vista del diseño, la implementación y la edición, mientras que la gestión

de DPB a nivel de imagen puede ser más óptima en los requisitos de memoria, al menos para algunos escenarios no típicos.

SUMARIO

5

[0007] En general, esta divulgación describe técnicas de codificación de vídeo. En particular, las técnicas están relacionadas con la codificación de vídeo y las técnicas de señalización que pueden aplicarse a los búferes de imágenes subdescodificadas (sub-DPB). En uno o más ejemplos de la divulgación, se describen técnicas para señalar parámetros de DPB y sub-DPB para una o más capas y/o una o más subcapas. Al utilizar las técnicas de la divulgación, la eficacia de la señalización puede aumentarse para flujos de bits de vídeo multicapa que incluyen una o más subcapas.

10

[0008] La invención está definida por las reivindicaciones independientes que forman parte de esta divulgación.

15

[0009] En un ejemplo de la divulgación, un procedimiento de descodificación de datos de vídeo comprende recibir un flujo de bits de vídeo codificado que incluye una pluralidad de imágenes, almacenar la pluralidad de imágenes en los uno o más sub-DPB, recibir un conjunto respectivo de parámetros de sub-DPB para cada punto de operación respectivo del flujo de bits de vídeo codificado, aplicar el conjunto respectivo de parámetros de sub-DPB a todas las capas de un conjunto de capas de salida para cada punto de operación respectivo, y realizar un proceso de gestión de sub-DPB en los uno o más sub-DPB de acuerdo con los conjuntos individuales respectivos de parámetros de sub-DPB recibidos.

20

[0010] En otro ejemplo de la divulgación, un aparato configurado para descodificar datos de vídeo comprende una o más sub-DPB configuradas para almacenar datos de vídeo, y un descodificador de vídeo configurado para recibir un flujo de bits de vídeo codificado que incluye una pluralidad de imágenes, almacenar la pluralidad de imágenes en los uno o más sub-DPB, recibir un conjunto respectivo de parámetros de sub-DPB para cada punto de operación respectivo del flujo de bits de vídeo codificado, aplicar el conjunto respectivo de parámetros de sub-DPB a todas las capas de un conjunto de capas de salida para cada punto de operación respectivo y realizar un proceso de gestión de sub-DPB en los uno o más sub-DPB de acuerdo con los respectivos conjuntos individuales de parámetros de sub-DPB recibidos.

25

30

[0011] En otro ejemplo de la divulgación, un aparato configurado para descodificar datos de vídeo comprende medios para recibir un flujo de bits de vídeo codificado que incluye una pluralidad de imágenes, medios para almacenar la pluralidad de imágenes en los uno o más sub-DPB, medios para recibir un conjunto respectivo de parámetros de sub-DPB para cada punto de operación respectivo del flujo de bits de vídeo codificado, medios para aplicar el conjunto respectivo de parámetros de sub-DPB a todas las capas de un conjunto de capas de salida para cada punto de operación respectivo, y medios para realizar un proceso de gestión de sub-DPB en los uno o más sub-DPB de acuerdo con los respectivos conjuntos individuales de parámetros de sub-DPB recibidos.

35

[0012] En otro ejemplo, esta descripción describe, un medio de almacenamiento legible por ordenador que almacena instrucciones que, cuando se ejecutan, hacen que uno o más procesadores de un dispositivo configurado para descodificar datos de vídeo reciban un flujo de bits de vídeo codificado que incluye una pluralidad de imágenes, almacenar la pluralidad de imágenes en los uno o más sub-DPB, recibir un conjunto respectivo de parámetros de sub-DPB para cada punto de operación respectivo del flujo de bits de vídeo codificado, aplicar el conjunto respectivo de parámetros de sub-DPB a todas las capas de un conjunto de capas de salida para cada punto de operación respectivo y realizar un proceso de gestión de sub-DPB en los uno o más sub-DPB de acuerdo con los conjuntos individuales respectivos de parámetros de sub-DPB recibidos.

45

[0013] En otro ejemplo de la divulgación, un procedimiento para codificar datos de vídeo comprende reconstruir una pluralidad de imágenes de un flujo de bits de vídeo codificado, almacenar la pluralidad de imágenes en los uno o más sub-DPB, generar un conjunto respectivo de parámetros de sub-DPB para cada punto de operación respectivo del flujo de bits de vídeo codificado, aplicar el conjunto respectivo de parámetros de sub-DPB a todas las capas de un conjunto de capas de salida para cada punto de operación respectivo, realizar un proceso de gestión de sub-DPB en los uno o más sub-DPB de acuerdo con los conjuntos individuales respectivos de parámetros de sub-DPB generados y señalar el conjunto respectivo de parámetros de sub-DPB en el flujo de bits de vídeo codificado.

50

55

[0014] En otro ejemplo de la divulgación, un aparato configurado para codificar datos de vídeo comprende uno o más sub-DPB configuradas para almacenar datos de vídeo, y un codificador de vídeo configurado para reconstruir una pluralidad de imágenes de un flujo de bits de vídeo codificado, almacenar la pluralidad de imágenes en los uno o más sub-DPB, generar un conjunto respectivo de parámetros de sub-DPB para cada punto de operación respectivo del flujo de bits de vídeo codificado, aplicar el conjunto respectivo de parámetros de sub-DPB a todas las capas de un conjunto de capas de salida para cada punto de operación respectivo, realizar un proceso de gestión de sub-DPB en los uno o más sub-DPB de acuerdo con los conjuntos individuales respectivos de parámetros de sub-DPB generados, y señalar el conjunto respectivo de parámetros de sub-DPB en el flujo de bits de vídeo codificado.

60

65

[0015] En otro ejemplo de la divulgación, un aparato configurado para codificar datos de vídeo comprende medios para reconstruir una pluralidad de imágenes de un flujo de bits de vídeo codificado, medios para almacenar la pluralidad de imágenes en los uno o más sub-DPB, medios para generar un conjunto respectivo de parámetros de sub-DPB para cada punto de operación respectivo del flujo de bits de vídeo codificado, medios para aplicar el conjunto respectivo de parámetros de sub-DPB a todas las capas de un conjunto de capas de salida para cada punto de operación respectivo, medios para realizar un proceso de gestión de sub-DPB en los uno o más sub-DPB de acuerdo con los conjuntos individuales respectivos de parámetros de sub-DPB generados, y medios para señalar el conjunto respectivo de parámetros de sub-DPB en el flujo de bits de vídeo codificado.

[0016] En otro ejemplo, esta divulgación describe un medio de almacenamiento legible por ordenador que almacena instrucciones que, cuando se ejecutan, hacen que uno o más procesadores de un dispositivo configurado para codificar datos de vídeo reconstruyan una pluralidad de imágenes de un flujo de bits de vídeo codificado que almacena la pluralidad de imágenes en los uno o más sub-DPB, generan un conjunto respectivo de parámetros de sub-DPB para cada punto de operación respectivo del flujo de bits de vídeo codificado, aplican el conjunto respectivo de parámetros de sub-DPB a todas las capas de un conjunto de capas de salida para cada punto de operación respectivo, realizan un proceso de gestión de sub-DPB en los uno o más sub-DPB de acuerdo con los conjuntos individuales respectivos de parámetros de sub-DPB generados, y señalizan el conjunto respectivo de parámetros de sub-DPB en el flujo de bits de vídeo codificado.

[0017] Los detalles de uno o más ejemplos se exponen en los dibujos adjuntos y en la descripción siguiente. Otras características, objetos y ventajas resultarán evidentes a partir de la descripción y de los dibujos, y a partir de las reivindicaciones.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

[0018]

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

La FIG. 2 es un diagrama conceptual que ilustra un orden de descodificación multivista de ejemplo.

La FIG. 3 es un diagrama conceptual que ilustra una estructura de predicción de ejemplo para la codificación multivista.

La FIG. 4 es un diagrama conceptual que ilustra capas de codificación de vídeo ajustables a escala de ejemplo.

La FIG. 5 es un diagrama conceptual que muestra un ejemplo de búfer de imágenes descodificadas (DPB) de ejemplo de acuerdo con ejemplos de la divulgación.

La FIG. 6 es un diagrama de bloques que ilustra un codificador de vídeo de ejemplo que puede implementar las técnicas descritas en esta divulgación.

La FIG. 7 es un diagrama de bloques que ilustra un descodificador de vídeo de ejemplo que puede implementar las técnicas descritas en esta divulgación.

La FIG. 8 es un diagrama de flujo que muestra un procedimiento de codificación de ejemplo de acuerdo con un ejemplo de las técnicas de esta divulgación.

La FIG. 9 es un diagrama de flujo que muestra un procedimiento de descodificación de ejemplo de acuerdo con las técnicas de esta divulgación.

DESCRIPCIÓN DETALLADA

[0019] En general, esta divulgación describe técnicas para la señalización de parámetros relacionados con búferes de imágenes descodificadas (DPB) en la codificación de vídeo multicapa. En particular, esta divulgación describe diversas técnicas para señalar parámetros relacionados con la gestión de subunidades de un DPB (también denominadas sub-DPB). Como se describirá con más detalle a continuación, las técnicas de esta divulgación incluyen técnicas para señalar parámetros de sub-DPB para una o más capas y una o más subcapas de un flujo de bits de vídeo multicapa. Al utilizar las técnicas de la divulgación, la eficacia de la señalización puede aumentarse para flujos de bits de vídeo multicapa que incluyen una o más subcapas.

[0020] En el contexto de esta divulgación, una capa puede ser una capa base o una o más capas de mejora en un proceso de codificación de vídeo ajustable a escala. Los procesos de codificación de vídeo ajustables a escala de ejemplo incluyen H.264/SVC (codificación de vídeo ajustable a escala) y las extensiones ajustables a escala de

la norma de codificación de vídeo de alta eficacia (HEVC), como HEVC ajustable a escala (SHVC). Además, una capa puede referirse a una o más vistas de textura en la codificación de vídeo en 3D o multivista. Además, una vista de profundidad en la codificación de vídeo en 3D también puede considerarse una capa. Para mencionar otro ejemplo, una capa puede corresponder a una vista única que incluye componentes de textura de la vista y componentes de profundidad de la vista. Los procesos de codificación multivista de ejemplo incluyen H.264/MVC (codificación multivista) y extensiones multivista de la norma HEVC, tal como HEVC multivista (MV-HEVC). Más en general, una capa puede referirse a un conjunto de unidades de capa de abstracción de red (NAL) que tienen la misma identificación de capa (por ejemplo, id de capa nuh).

[0021] Por lo tanto, el término "capa" se usa en general en esta divulgación para referirse a las capas de base y/o de mejora en el sentido de codificación de vídeo ajustable a escala, o vistas (incluyendo componentes de textura de la vista y/o componentes de profundidad de la vista) en el sentido de codificación de vídeo multivista y 3D. Por consiguiente, el término codificación de vídeo multicapa en general se refiere a técnicas de codificación de vídeo ajustables a escala, técnicas de codificación de vídeo multivista y técnicas de codificación de vídeo en 3D, incluyendo codificación multivista más profundidad. Las técnicas de esta divulgación pueden aplicarse a cualquiera de estos escenarios de codificación de vídeo, incluyendo las extensiones multivista, extensiones de vídeo en 3D y extensiones ajustables a escala de la HEVC y/o H.264/AVC.

[0022] En el contexto de esta divulgación, el término "subcapa" se refiere a una capa temporal ajustable a escala de un flujo de bits temporal ajustable a escala. Es decir, un flujo de bits en el que una o más subcapas diferentes pueden descodificarse para emitir vídeo a diferentes frecuencias de trama. Una capa puede incluir una o más subcapas.

[0023] Las técnicas que se describen a continuación se pueden aplicar a las técnicas de codificación de vídeo ajustable a escala, multivista y en 3D basadas en códecs avanzados, incluyendo la codificación de dos o más vistas de una imagen con mapas de profundidad. Entre las normas de codificación de vídeo se incluyen ITU-T H.261, ISO/IEC MPEG-1 Visual, ITU-T H.262 o ISO/IEC MPEG-2 Visual, ITU-T H.263, ISO/IEC MPEG-4 Visual e ITU-T H.264 (también conocida como ISO/IEC MPEG-4 AVC), incluidas sus extensiones de codificación de vídeo ajustable a escala (SVC) y de codificación de vídeo multivista (MVC). Además, una nueva norma de codificación de vídeo, concretamente, la Codificación de Vídeo de Alta Eficacia (HEVC), ha sido desarrollada por el Equipo de Colaboración Conjunta en Codificación de Vídeo (JCT-VC) del Grupo de Expertos en Codificación de Vídeo (VCEG) de la ITU-T y el Grupo de Expertos en Imágenes en Movimiento (MPEG) de ISO/IEC. La norma HEVC se describe en ITU-T H.265, SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS, Infrastructure of Audiovisual Services-Coding of Moving Video, "High Efficiency Video Coding," Abril de 2013 (en lo sucesivo, "HEVC").

[0024] Se han propuesto diversas extensiones de HEVC. Una de dichas extensiones es la extensión del alcance de HEVC, que se describe en "High Efficiency Video Coding (HEVC) Range Extensions text specification: Draft 4," JCTVC-N1005_v1, agosto de 2013 (en lo sucesivo, "JCTVC-N1005"). Un borrador de trabajo (WD) reciente de la norma HEVC ajustable a escala (SHVC), titulado "High efficiency video coding (HEVC) scalable extension draft 3", Equipo de colaboración conjunta sobre codificación de vídeo (JCT-VC) de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 14ª reunión: Viena, AT, 25 de julio - 2 de agosto de 2013, y denominado SHVC WD3 en lo sucesivo, está disponible en http://phenix.it-sudparis.eu/jct/doc_end_user/documents/14_Vienna/wg11/JCTVC-N1008-v3.zip. El JCT-3V también está desarrollando la extensión multivista de HEVC, concretamente, la MV-HEVC. Un borrador de trabajo reciente de la MV-HEVC (en lo sucesivo WD4), está disponible en http://phenix.it-sudparis.eu/jct2/doc_end_user/documents/4_Incheon/wg11/JCT3V-D1004-v3.zip. Un borrador de trabajo más reciente de MV-HEVC, titulado "MV-HEVC Draft Text 8", Equipo de Colaboración Conjunta sobre Extensiones de Codificación de Vídeo en 3D de ITU-T SG 16 WP3 e ISO/IEC JTC 1/SC 29/WG11, 8ª reunión: Valencia, ES, 29 de marzo - 4 de abril de 2014, está disponible en http://phenix.it-sudparis.eu/jct3v/doc_end_user/documents/8_Valencia/wg11/JCT3V-H0002-v1.zip.

[0025] La FIG. 1 es un diagrama de bloques que ilustra un sistema de codificación y descodificación de vídeo de ejemplo puede utilizar las técnicas para la gestión y señalización del búfer de imágenes descodificadas (DPB) en un proceso de codificación de vídeo multicapa descrito en esta divulgación. Como se muestra en la FIG. 1, el sistema 10 incluye un dispositivo de origen 12 que genera datos de vídeo codificados que serán descodificados en un momento posterior por un dispositivo de destino 14. El dispositivo de origen 12 y el dispositivo de destino 14 pueden comprender cualquiera entre una amplia gama de dispositivos, incluyendo ordenadores de mesa, ordenadores plegables (es decir, portátiles), tabletas electrónicas, cajas de conexión, aparato telefónico con microteléfono tales como los denominados teléfonos "inteligentes", los denominados paneles "inteligentes", televisores, cámaras, dispositivos de visualización, reproductores de medios digitales, consolas de videojuegos, dispositivos de emisión de vídeo en tiempo real o similares. En algunos casos, el dispositivo de origen 12 y el dispositivo de destino 14 pueden estar equipados para la comunicación inalámbrica.

[0026] El dispositivo de destino 14 puede recibir los datos de vídeo codificados que se van a descodificar por medio de un enlace 16. El enlace 16 puede comprender cualquier tipo de medio o dispositivo capaz de transferir los datos de vídeo codificados desde el dispositivo de origen 12 hasta el dispositivo de destino 14. En un ejemplo, el enlace 16 puede comprender un medio de comunicación para permitir que el dispositivo de origen 12 transmita

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

[0027] De forma alternativa, los datos codificados se pueden emitir desde la interfaz de salida 22 a un dispositivo de almacenamiento 34. De forma similar, una interfaz de entrada puede acceder a los datos codificados del dispositivo de almacenamiento 34. El dispositivo de almacenamiento 34 puede incluir cualquiera de una variedad de medios de almacenamiento de datos de acceso distribuido o local, tales como una unidad de disco duro, discos Blu-ray, DVD, CD-ROM, una memoria flash, una memoria volátil o no volátil o cualquier otro medio de almacenamiento digital adecuado para almacenar datos de vídeo codificados. En otro ejemplo, el dispositivo de almacenamiento 34 puede corresponder a un servidor de archivos o a otro dispositivo de almacenamiento intermedio que puede contener el vídeo codificado generado por el dispositivo de origen 12. El dispositivo de destino 14 puede acceder a datos de vídeo almacenados en el dispositivo de almacenamiento 34 a través de transmisión continua 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 (por ejemplo, para un sitio web), un servidor FTP, dispositivos de almacenamiento conectados en red (NAS) o una unidad de disco local. El dispositivo de destino 14 puede acceder a los datos de vídeo codificados a través de cualquier conexión de datos estándar, incluida una conexión a Internet. Esto puede incluir un canal inalámbrico (por ejemplo, una conexión Wi-Fi), una conexión alámbrica (por ejemplo, DSL, módem de cable, etc.) o una combinación de ambas que sea adecuada 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 34 puede ser una transmisión continua, una transmisión de descarga o una combinación de ambas cosas.

[0028] Las técnicas de esta divulgación para la gestión y señalización de DPB en un proceso de codificación de vídeo multicapa, como SHVC, MV-HEVC o 3D-HEVC como ejemplos, no se limitan necesariamente a aplicaciones o ajustes inalámbricos. Las técnicas se pueden aplicar a la codificación de vídeo como apoyo a cualquiera de una variedad de aplicaciones multimedia, tales como radiodifusiones de televisión por aire, transmisiones de televisión por cable, transmisiones de televisión por satélite, transmisiones continuas de vídeo, por ejemplo, a través de Internet, codificación de vídeo digital para su almacenamiento en un medio de almacenamiento de datos, descodificación de vídeo digital almacenado en un medio de almacenamiento de datos, u otras aplicaciones. En algunos ejemplos, el sistema 10 puede estar configurado para prestar soporte a una transmisión de vídeo unidireccional o bidireccional, a fin de prestar soporte a aplicaciones tales como la transmisión continua de vídeo, la reproducción de vídeo, la radiodifusión de vídeo y/o la videotelefonía.

[0029] En el ejemplo de la FIG. 1, el dispositivo de origen 12 incluye una fuente de vídeo 18, un codificador de vídeo 20 y una interfaz de salida 22. En algunos casos, la interfaz de salida 22 puede incluir un modulador/desmodulador (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 captación de vídeo, por ejemplo, una videocámara, un archivo de vídeo que contiene vídeo captado previamente, una interfaz de alimentación de vídeo para recibir vídeo desde un proveedor de contenido de vídeo y/o un sistema de gráficos por ordenador para generar datos de gráficos por ordenador como el vídeo de origen, o una combinación de dichas fuentes. A modo de ejemplo, si la fuente de vídeo 18 es una videocámara, el dispositivo de origen 12 y el dispositivo de destino 14 pueden formar los denominados teléfonos con cámara o videoteléfonos. Sin embargo, las técnicas descritas en esta divulgación pueden ser aplicables a la codificación de vídeo en general, y se pueden aplicar a aplicaciones inalámbricas y/o cableadas.

[0030] El vídeo captado, precaptado o generado por ordenador se puede codificar por el codificador de vídeo 20. Los datos de vídeo codificados se pueden transmitir directamente al dispositivo de destino 14 por medio de la interfaz de salida 22 del dispositivo de origen 12. Los datos de vídeo codificados se pueden almacenar de forma adicional (o alternativa) en el dispositivo de almacenamiento 34 para un posterior acceso por el dispositivo de destino 14 u otros dispositivos, para su descodificación y/o reproducción.

[0031] El dispositivo de destino 14 incluye una interfaz de entrada 28, un descodificador de vídeo 30 y un dispositivo de visualización 32. En algunos casos, la interfaz de entrada 28 puede incluir un receptor y/o un módem. La interfaz de entrada 28 del dispositivo de destino 14 recibe los datos de vídeo codificados a través del enlace 16. Los datos de vídeo codificados transmitidos a través del enlace 16, o proporcionados en el dispositivo de almacenamiento 34, pueden incluir una diversidad de elementos sintácticos generados por el codificador de vídeo 20 para su uso por un descodificador de vídeo, tal como el descodificador de vídeo 30, en la descodificación de los datos de vídeo. Dichos elementos sintácticos se pueden incluir 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.

[0032] El dispositivo de visualización 32 se puede integrar con, o ser externo a, el dispositivo de destino 14. En algunos ejemplos, el dispositivo de destino 14 puede incluir un dispositivo de visualización integrado y también estar configurado para interconectarse con un dispositivo de visualización externo. En otros ejemplos, el dispositivo de destino 14 puede ser un dispositivo de visualización. En general, el dispositivo de visualización 32 visualiza los datos de vídeo descodificados ante un usuario, y puede comprender cualquiera de una variedad de dispositivos de visualización, tales como una pantalla de cristal líquido (LCD), una pantalla de plasma, una pantalla de diodos orgánicos emisores de luz (OLED) u otro tipo de dispositivo de visualización.

[0033] El codificador de vídeo 20 y el descodificador de vídeo 30 pueden funcionar de acuerdo con una norma de codificación de vídeo, tal como la norma de codificación de vídeo de alta eficacia (HEVC), y pueden ajustarse al modelo de prueba de HEVC (HM). En particular, en algunos ejemplos, el codificador de vídeo 20 y el descodificador de vídeo pueden operar de acuerdo con extensiones de la HEVC que admiten codificación de vídeo ajustable a escala, multivista y/o multivista más profundidad. En otros ejemplos, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden operar de acuerdo con otras normas de propiedad o industriales, tales como la norma ITU-T H.264, de forma alternativa denominada MPEG-4, parte 10, codificación avanzada de vídeo (AVC), o extensiones de dichas normas, incluyendo H.264/SVC. Sin embargo, las técnicas de esta divulgación no están limitadas a ninguna norma de codificación particular. Otros ejemplos de normas de compresión de vídeo incluyen MPEG-2 e ITU-T H.263. En particular, de acuerdo con las técnicas de esta divulgación, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden operar de acuerdo con una norma de codificación de vídeo capaz de codificación ajustable a escala (por ejemplo, SHVC) y/o multivista (por ejemplo, MV-HEVC o 3D-HEVC).

[0034] Aunque no se muestra en la FIG. 1, en algunos aspectos, tanto el codificador de vídeo 20 como el descodificador de vídeo 30 pueden estar integrados con un codificador y descodificador de audio, y pueden incluir unidades MUX-DEMUX adecuadas, u otro tipo de hardware y software, para gestionar la codificación tanto de audio como de vídeo en un flujo de datos común o en flujos de datos separados. Si procede, en algunos ejemplos, las unidades MUX-DEMUX pueden ajustarse al protocolo de multiplexor ITU H.223 o a otros protocolos, tales como el protocolo de datagramas de usuario (UDP).

[0035] Tanto el codificador de vídeo 20 como el descodificador de vídeo 30 se pueden implementar como cualquiera de una variedad de circuitos codificadores adecuados, tales como uno o más microprocesadores, procesadores de señales digitales (DSP), circuitos integrados específicos de la aplicación (ASIC), matrices de puertas programables in situ (FPGA), lógica discreta, software, hardware, firmware o cualquier combinación de los mismos. Cuando las técnicas se implementan parcialmente en software, un dispositivo puede almacenar instrucciones para el software en un medio adecuado no transitorio legible por ordenador, y ejecutar las instrucciones en hardware usando uno o más procesadores para realizar las técnicas de esta divulgación. Tanto el codificador de vídeo 20 como el descodificador de vídeo 30 se pueden incluir en uno o más codificadores o descodificadores, cualquiera de los cuales se puede integrar como parte de un codificador/descodificador (CÓDEC) combinado en un dispositivo respectivo.

[0036] La siguiente sección de la divulgación proporcionará un antecedente de la norma HEVC. Los trabajos de estandarización HEVC se basaron en un modelo en evolución de un dispositivo de codificación de vídeo denominado modelo de prueba HEVC (HM). El HM supone varias capacidades adicionales de los dispositivos de codificación de vídeo respecto a los dispositivos existentes, de acuerdo con, por ejemplo, la norma ITU-T H.264/AVC. Por ejemplo, mientras que la norma H.264 proporciona nueve modos de codificación mediante intrapredicción, el HM puede proporcionar hasta treinta y tres modos de codificación mediante intrapredicción.

[0037] En general, el modelo de trabajo del HM describía que una trama o imagen de vídeo se puede dividir en una secuencia de bloques de árbol (también denominados bloques de árbol de codificación (CTB) o unidades de codificación más grandes (LCU)), que incluyen tanto muestras de luma como de croma. Un bloque de árbol tiene un propósito similar al de un macrobloque de la norma H.264. Un fragmento incluye un número de bloques de árbol consecutivos en orden de codificación. Una trama o imagen de vídeo puede dividirse en uno o más fragmentos. Cada bloque de árbol puede dividirse en unidades de codificación (CU) de acuerdo con un árbol cuaternario. Por ejemplo, un bloque de árbol, tal como un nodo raíz del árbol cuaternario, puede dividirse en cuatro nodos hijo, y cada nodo hijo puede, a su vez, ser un nodo padre y dividirse en otros cuatro nodos hijo. Un nodo hijo final, no dividido, tal como un nodo hoja del árbol cuaternario, comprende un nodo de codificación, es decir, un bloque de vídeo codificado. Los datos sintácticos asociados a un flujo de bits codificado pueden definir un número máximo de veces que puede dividirse un bloque de árbol, y también pueden definir un tamaño mínimo de los nodos de codificación.

[0038] Una CU incluye un nodo de codificación y unidades de predicción (PU) y unidades de transformada (TU) asociadas al nodo de codificación. Un tamaño de la CU corresponde, en general, a un tamaño del nodo de codificación y debe ser típicamente de forma cuadrada. El tamaño de la CU puede variar desde 8x8 píxeles hasta el tamaño del bloque de árbol, con un máximo de 64x64 píxeles o más. Cada CU puede contener una o más PU y una o más TU. Los datos sintácticos asociados a una CU pueden describir, por ejemplo, la división de la CU en una o más PU. Los modos de división pueden diferir entre si la CU está codificada en modo directo o de salto,

codificada en modo de intrapredicción o codificada en modo de interpredicción. Las PU se pueden dividir para tener una forma no cuadrada. Los datos sintácticos asociados a una CU también pueden describir, por ejemplo, la división de la CU en una o más TU de acuerdo con un árbol cuaternario. Una TU puede tener una forma cuadrada o no cuadrada.

5

[0039] La norma HEVC admite transformaciones de acuerdo con las TU, que pueden ser diferentes para CU diferentes. El tamaño de las TU se basa típicamente en el tamaño de las PU dentro de una CU dada definida para una LCU dividida, aunque puede que no sea siempre así. Las TU son típicamente del mismo tamaño o de un tamaño más pequeño que las PU. En algunos ejemplos, las muestras residuales correspondientes a una CU pueden subdividirse en unidades más pequeñas usando una estructura de árbol cuaternario conocida como "árbol cuaternario residual" (RQT). Los nodos hoja del RQT pueden denominarse unidades de transformada (TU). Los valores de diferencias de píxeles asociados a las TU pueden transformarse para producir coeficientes de transformada, que pueden cuantificarse.

10

[0040] En general, una PU incluye datos relacionados con el procedimiento de predicción. Por ejemplo, cuando la PU se codifica en intramodo, la PU puede incluir datos que describen un modo de intrapredicción para la PU. Para mencionar otro ejemplo, cuando la PU se codifica en 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 (por ejemplo, precisión de un cuarto de píxel o precisión de un octavo de píxel), una imagen de referencia a la que apunta el vector de movimiento y/o una lista de imágenes de referencia (por ejemplo, la Lista 0, la Lista 1 o la Lista C) para el vector de movimiento, que puede indicarse mediante una dirección de predicción.

20

[0041] En general, se usa una TU para los procedimientos de transformada y cuantificación. Una CU dada que tiene una o más PU también puede incluir una o más unidades de transformada (TU). Después de la predicción, el codificador de vídeo 20 puede calcular unos valores residuales a partir del bloque de vídeo identificado por el nodo de codificación de acuerdo con la PU. El nodo de codificación se actualiza luego para hacer referencia a los valores residuales, en lugar del bloque de vídeo original. Los valores residuales comprenden valores de diferencias de píxeles que pueden transformarse en coeficientes de transformada, cuantificarse y explorarse usando las transformadas y otra información de transformada especificada en las TU, para producir coeficientes de transformada en serie para la codificación por entropía. El nodo de codificación puede, una vez más, actualizarse para referirse a estos coeficientes de transformada en serie. Esta divulgación usa típicamente el término "bloque de vídeo" para referirse a un nodo 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, es decir, CTB, LCU o una CU, que incluye un nodo de codificación y unas PU y TU.

25

30

35

[0042] Una secuencia de vídeo incluye típicamente una serie de tramas o imágenes de vídeo. Un grupo de imágenes (GOP) comprende, en general, una serie de una o más de las imágenes de vídeo. Un GOP puede incluir datos sintácticos en una cabecera del GOP, una cabecera de una o más de las imágenes, o en otras ubicaciones, que describen una serie de imágenes incluidas en el GOP. Cada fragmento de una imagen puede incluir datos sintácticos de fragmento que describen un modo de codificación para el fragmento respectivo. El codificador de vídeo 20 actúa típicamente en bloques de vídeo dentro de fragmentos de vídeo individuales a fin de 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 una norma de codificación especificada.

40

45

[0043] En un ejemplo, el HM admite la predicción en diversos tamaños de PU. Si se supone que el tamaño de una CU particular es $2N \times 2N$, el HM admite la intrapredicción en tamaños de PU de $2N \times 2N$ o $N \times N$, así como la interpredicción en tamaños de PU simétricas de $2N \times 2N$, $2N \times N$, $N \times 2N$ o $N \times N$. El HM también admite la división asimétrica para la interpredicción en tamaños de PU de $2N \times nU$, $2N \times nD$, $nL \times 2N$ y $nR \times 2N$. En la división asimétrica, una dirección de una CU no está dividida, mientras que la otra dirección está dividida entre un 25 % y un 75 %. La parte de la CU correspondiente a la partición del 25 % está indicada por una "n" seguida de una indicación de "Arriba", "Abajo", "Izquierda" o "Derecha". Así, por ejemplo, " $2N \times nU$ " se refiere a una CU $2N \times 2N$ que está dividida horizontalmente con una PU $2N \times 0,5N$ encima y una PU $2N \times 1,5N$ debajo.

50

55

[0044] En esta divulgación, " $N \times N$ " y " N por N " pueden usarse indistintamente para hacer referencia a las dimensiones de píxeles de un bloque de vídeo en términos de dimensiones verticales y horizontales, por ejemplo, 16×16 píxeles o 16 por 16 píxeles. En general, un bloque de 16×16 tendrá 16 píxeles en una dirección vertical ($y = 16$) y 16 píxeles en una dirección horizontal ($x = 16$). Asimismo, un bloque de tamaño $N \times N$ tiene, en general, N píxeles en una dirección vertical y N píxeles en una dirección horizontal, donde N representa un valor entero no negativo. Los píxeles de un bloque se pueden disponer en filas y columnas. Además, no es necesario que los bloques tengan necesariamente el mismo número de píxeles en la dirección horizontal que en la dirección vertical. Por ejemplo, los bloques pueden comprender $N \times M$ píxeles, donde M no es necesariamente igual a N .

60

65

[0045] Tras la codificación intrapredictiva o interpredictiva, usando las PU de una CU, el codificador de vídeo 20 puede calcular datos residuales a los que se aplican las transformadas especificadas por las TU de la CU. Los datos residuales pueden corresponder a diferencias de píxeles entre píxeles de la imagen no codificada y valores de predicción correspondientes a las CU. El codificador de vídeo 20 puede formar los datos residuales para la CU, y a continuación transformar los datos residuales para producir coeficientes de transformada.

[0046] Seguidamente a cualquier transformada para producir coeficientes de transformada, el codificador de vídeo 20 puede realizar la cuantificación de los coeficientes de transformada. La cuantificación se refiere, en general, a un proceso en el que unos coeficientes de transformada se cuantifican para reducir posiblemente la cantidad de datos usados para representar los coeficientes, proporcionando una compresión adicional. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos, o a la totalidad, de los coeficientes. Por ejemplo, un valor de n bits se puede redondear a la baja hasta un valor de m bits durante la cuantificación, donde n es mayor que m .

[0047] En algunos ejemplos, el codificador de vídeo 20 puede utilizar un orden de exploración predefinido para explorar los coeficientes de transformada cuantificados, para producir un vector en serie que se pueda codificar por entropía. En otros ejemplos, el codificador de vídeo 20 puede realizar una exploración adaptativa. Después de explorar los coeficientes de transformada cuantificados para formar un vector unidimensional, el codificador de vídeo 20 puede realizar la codificación por entropía del vector unidimensional, por ejemplo, de acuerdo con la codificación aritmética binaria adaptativa al contexto (CABAC) u otra metodología de codificación por entropía. Para realizar la CABAC, el codificador de vídeo 20 puede asignar un contexto, dentro de un modelo de contexto, a un símbolo que se va a transmitir. El contexto se puede referir, por ejemplo, a si los valores vecinos del símbolo son distintos de cero o no. El codificador de vídeo 20 también puede realizar la codificación por entropía de los elementos sintácticos asociados a los datos de vídeo codificados, para su uso por el decodificador de vídeo 30 en la decodificación de los datos de vídeo.

[0048] Las extensiones HEVC también se están desarrollando actualmente en el JCT-3V y JCT-VC. En JCT-3V, se están desarrollando dos extensiones HEVC relacionadas con multivista, la extensión multivista (MV-HEVC) y la extensión de vídeo en 3D (3D-HEVC). Además, se están desarrollando dos extensiones AVC, la MVC+D y la 3D-AVC.

[0049] Versiones de ejemplo de las normas en curso se enumeran de la siguiente manera:

- T. Suzuki, M. M. Hannuksela, Y. Chen, S. Hattori, G. Sullivan, "MVC Extension for Inclusion of Depth Maps Draft Text 6", JCT3V-C1001, Equipo de Colaboración Conjunta sobre Extensiones de Codificación de Vídeo en 3D de ITU-T SG 16 WP3 e ISO/IEC JTC 1/SC 29/WG 11, 4ª reunión, disponible en http://phenix.int-evry.fr/jct2/doc_end_user/documents/3_Geneva/wg11/JCT3V-C1001-v3.zip.

- M. M. Hannuksela, Y. Chen, T. Suzuki, J.-R. Ohm, G. Sullivan, "3D-AVC Draft Text 8", JCT3V-F1002, Equipo de Colaboración Conjunta sobre Extensiones de Codificación de Vídeo en 3D de ITU-T SG 16 WP3 e ISO/IEC JTC 1/SC 29/WG 11, 6ª reunión, disponible en http://phenix.int-evry.fr/jct2/doc_end_user/documents/6_Geneva/wg11/JCT3V-F1002-v3.

- JCT3V-F1004, "MV-HEVC Draft Text 6", G. Tech, K. Wegner, Y. Chen, M. Hannuksela, Equipo de Colaboración Conjunta sobre el Desarrollo de Extensiones de Codificación de Vídeo en 3D de ITU-T SG 16 WP3 e ISO/IEC JTC 1/SC 29/WG 11, 6ª reunión, disponible en http://phenix.int-evry.fr/jct2/doc_end_user/documents/6_Geneva/wg11/JCT3V-F1004-v6.zip.

- Gerhard Tech, Krzysztof Wegner, Ying Chen, Sehoon Yea, "3D-HEVC Draft Text 2", JCT3V-F1001, Equipo de Colaboración Conjunta sobre el Desarrollo de Extensiones de Codificación de Vídeo en 3D de ITU-T SG 16 WP3 e ISO/IEC JTC 1/SC 29/WG 11, 6ª reunión, disponible en http://phenix.int-evry.fr/jct2/doc_end_user/documents/6_Geneva/wg11/JCT3V-F1001-v2.

[0050] Ahora se analizarán las técnicas de codificación de vídeo multivista de extensiones de la norma H.264/codificación avanzada de vídeo (AVC). Sin embargo, las técnicas de esta divulgación pueden ser aplicables con cualquier estándar de codificación de vídeo que admita la codificación multivista y/o las propuestas multivista de codificación en 3D para la norma HEVC emergente (por ejemplo, MV-HEVC y 3D-HEVC).

[0051] La codificación de vídeo multivista (MVC) es una extensión de H.264/AVC. En la FIG. 2 se muestra un orden de decodificación MVC típico (es decir, un orden de flujo de bits). La disposición del orden de decodificación se denomina codificación de tiempo en primer lugar. Obsérvese que el orden de decodificación de las unidades de acceso puede no ser idéntico al orden de salida o de visualización de las unidades de acceso. En la FIG. 2, S0 a S7 se refieren, cada uno, a diferentes vistas del vídeo multivista. Cada una entre T1 a T9 representa una instancia de tiempo de salida. Una unidad de acceso puede incluir las imágenes codificadas de todas las vistas para una instancia de tiempo de salida. Por ejemplo, una primera unidad de acceso puede incluir

todas las vistas S0 a S7 para la instancia de tiempo T1, una segunda unidad de acceso puede incluir todas las vistas S0 a S7 para la instancia de tiempo T2 y así sucesivamente.

[0052] En aras de la brevedad, la divulgación puede utilizar las siguientes definiciones:

componente de vista: Una *representación codificada* de una *vista* en una *única unidad de acceso*. Cuando una vista incluye tanto representaciones de textura como representaciones de profundidad codificadas, un componente de vista puede incluir un componente de textura de la vista y un componente de profundidad de la vista.

componente de textura de la vista: Una *representación codificada* de la textura de una vista en una *única unidad de acceso*.

componente de profundidad de la vista: Una *representación codificada* de la profundidad de una vista en una *única unidad de acceso*.

[0053] Como se ha analizado anteriormente, en el contexto de esta divulgación, tanto el componente de la vista, el componente de textura de la vista como el componente de profundidad de la vista pueden denominarse, en general, una capa. En la FIG. 2, cada una de las vistas incluye conjuntos de imágenes. Por ejemplo, la vista S0 incluye un conjunto de imágenes 0, 8, 16, 24, 32, 40, 48, 56 y 64, la vista S1 incluye el conjunto de imágenes 1, 9, 17, 25, 33, 41, 49, 57 y 65, y así sucesivamente. Cada conjunto incluye dos imágenes: una imagen se denomina componente de textura de la vista y la otra imagen se denomina componente de profundidad de la vista. El componente de textura de la vista y el componente de profundidad de la vista dentro de un conjunto de imágenes de una vista se pueden considerar como correspondientes entre sí. Por ejemplo, el componente de textura de la vista dentro de un conjunto de imágenes de una vista se considera correspondiente al componente de profundidad de la vista dentro del conjunto de las imágenes de la vista, y viceversa (es decir, el componente de profundidad de la vista corresponde a su componente de textura de la vista en el conjunto, y viceversa). Como se usa en esta descripción, un componente de textura de la vista que corresponde a un componente de profundidad de la vista puede considerarse que forma parte de una misma vista que el componente de profundidad en una única unidad de acceso.

[0054] El componente de textura de la vista incluye el contenido de la imagen real que se visualiza. Por ejemplo, el componente de textura de la vista puede incluir los componentes de luma (Y) y croma (Cb y Cr). El componente de profundidad de la vista puede indicar profundidades relativas de los píxeles en su componente de textura de la vista correspondiente. A modo de ejemplo, el componente de profundidad de la vista es una imagen en escala de grises que incluye solamente valores de luma. En otras palabras, el componente de profundidad de la vista puede que no transmita ningún contenido de imagen, sino que, en cambio, proporcione una medida de las profundidades relativas de los píxeles en el componente de textura de la vista.

[0055] Por ejemplo, un píxel blanco puro en el componente de profundidad de la vista indica que su correspondiente píxel, o correspondientes píxeles, en el correspondiente componente de textura de la vista, está(n) más cerca, desde la perspectiva del observador, y un píxel negro puro en el componente de profundidad de la vista indica que su correspondiente píxel, o correspondientes píxeles, en el correspondiente componente de textura de la vista, está(n) más lejos, desde la perspectiva del observador. Los diversos tonos de gris entre blanco y negro indican diferentes niveles de profundidad, de manera que un aumento en la oscuridad del tono de gris de un píxel en la vista de profundidad es indicativo de un aumento en el nivel de profundidad asociado con el píxel correspondiente en la vista de textura. Por ejemplo, un píxel muy gris en el componente de profundidad de la vista indica que su píxel correspondiente en el componente de textura de la vista está más alejado que un píxel ligeramente gris en el componente de profundidad de la vista. Dado que solamente es necesaria la escala de grises para identificar la profundidad de los píxeles, el componente de profundidad de la vista no necesita incluir componentes de croma, ya que los valores de color para el componente de profundidad de la vista pueden no tener ningún sentido.

[0056] El componente de profundidad de la vista que utiliza solo valores de luma (por ejemplo, valores de intensidad) para identificar la profundidad se proporciona con fines ilustrativos y no debería considerarse limitante. En otros ejemplos, se puede utilizar cualquier técnica para indicar las profundidades relativas de los píxeles en el componente de textura de la vista.

[0057] En la FIG. 3, una estructura de predicción MVC típica (que incluye tanto la predicción entre imágenes dentro de cada vista como la predicción entre vistas) para la codificación de vídeo multivista. La estructura de predicción de la FIG. 3 también se puede usar con MV-HEVC. Las direcciones de predicción se indican con flechas, el objeto al que se apunta utiliza el objeto desde el que se apunta como referencia de predicción. En la MVC, la predicción entre vistas se apoya en la compensación de movimiento por disparidad, que usa la sintaxis de la compensación de movimiento de la H.264/AVC, pero permite que una imagen en una vista diferente sea usada como imagen de referencia.

[0058] En el ejemplo de la FIG. 3, se ilustran ocho vistas (que tienen identificadores de vista "S0" a "S7") y se ilustran doce ubicaciones temporales ("T0" a "T11") para cada vista. Es decir, cada fila en la FIG. 3 corresponde a una vista, mientras que cada columna indica una ubicación temporal.

5 **[0059]** Aunque la MVC tiene una denominada vista base que es descodificable mediante los descodificadores de la H.264/AVC y los pares de vistas estéreo podrían ser admitidos también por la MVC, la ventaja de la MVC es que la MVC podría admitir un ejemplo que usa más de dos vistas como entrada de vídeo en 3D y descodifica este vídeo en 3D representado por las múltiples vistas. Un visualizador de un cliente que tiene un descodificador de MVC puede esperar contenido de vídeo en 3D con múltiples vistas.

10 **[0060]** Las imágenes en la FIG. 3 se indican en la intersección de cada fila y cada columna. La norma H.264/AVC puede usar el término trama para representar una porción del vídeo. Esta divulgación puede usar los términos imagen y trama indistintamente.

15 **[0061]** Las imágenes en la FIG. 3 se ilustran usando un bloque que incluye una letra, la letra que designa si la imagen correspondiente está intracodificada (es decir, es una imagen I), o intercodificada en una dirección (es decir, como una imagen P) o en múltiples direcciones (es decir, como una imagen B). En general, las predicciones se indican mediante flechas, donde las imágenes a las que se apunta usan imágenes desde las que se apunta como referencia de predicción. Por ejemplo, la imagen P de la vista S2 en la ubicación temporal T0 se predice a partir de la imagen I de la vista S0 en la ubicación temporal T0.

20 **[0062]** Al igual que con la codificación de vídeo de vista única, las imágenes de una secuencia de vídeo de codificación de vídeo multivista pueden codificarse de forma predictiva con respecto a imágenes en diferentes ubicaciones temporales. Por ejemplo, la imagen b de la vista S0 en la ubicación temporal T1 tiene una flecha que apunta a la misma desde la imagen I de la vista S0 en la ubicación temporal T0, indicando que la imagen b se predice a partir de la imagen I. Adicionalmente, sin embargo, en el contexto de la codificación de vídeo multivista, las imágenes se pueden predecir entre vistas. Es decir, un componente de la vista puede usar los componentes de la vista en otras vistas como referencia. En la MVC, por ejemplo, la predicción entre vistas se realiza como si el componente de la vista de otra vista fuera una referencia de interpredicción. Las posibles referencias entre vistas se señalizan en la extensión de la MVC del conjunto de parámetros de secuencia (SPS) y pueden ser modificadas por el proceso de construcción de la lista de imágenes de referencia, que habilita el ordenamiento flexible de las referencias de interpredicción o predicción entre vistas. La predicción entre vistas también es una característica de una extensión multivista propuesta de la HEVC, que incluye la 3D-HEVC (multivista más profundidad).

35 **[0063]** La FIG. 3 proporciona diversos ejemplos de predicción entre vistas. Las imágenes de la vista S1, en el ejemplo de la FIG. 3, se ilustran como predichas a partir de imágenes en diferentes ubicaciones temporales de la vista S1, así como predichas entre vistas a partir de imágenes de las vistas S0 y S2 en las mismas ubicaciones temporales. Por ejemplo, la imagen b de la vista S1 en la ubicación temporal T1 se predice a partir de cada una de las imágenes B de la vista S1 en las ubicaciones temporales T0 y T2, así como las imágenes b de las vistas S0 y S2 en la ubicación temporal T1.

40 **[0064]** En algunos ejemplos, la FIG. 3 puede verse como una ilustración de los componentes de textura de la vista. Por ejemplo, las imágenes I, P, B y b, ilustradas en la FIG. 3, se pueden considerar componentes de textura de la vista para cada una de las vistas. De acuerdo con las técnicas descritas en esta divulgación, para cada uno de los componentes de textura de la vista ilustrados en la FIG. 3 existe un componente de profundidad de la vista correspondiente. En algunos ejemplos, los componentes de profundidad de la vista se pueden predecir de una manera similar a la ilustrada en la FIG. 3 para los componentes de textura de la vista correspondientes.

45 **[0065]** La codificación de dos vistas también se puede admitir por la MVC. Una de las ventajas de la MVC es que un codificador de MVC puede tomar más de dos vistas como una entrada de vídeo en 3D y un descodificador de MVC puede descodificar dicha representación multivista. De tal modo, cualquier visualizador con un descodificador de MVC puede descodificar contenidos de vídeo en 3D con más de dos vistas.

50 **[0066]** Como se ha analizado anteriormente, en la MVC, se permite la predicción entre vistas entre imágenes en la misma unidad de acceso (lo que quiere decir, en algunos casos, con la misma instancia de tiempo). Cuando se codifica una imagen en una de las vistas no base, se puede añadir una imagen a una lista de imágenes de referencia, si está en una vista diferente pero dentro de una misma instancia de tiempo. Una imagen de referencia de predicción entre vistas se puede poner en cualquier posición de una lista de imágenes de referencia, como es el caso con cualquier imagen de interpredicción de referencia. Como se muestra en la FIG. 3, un componente de la vista puede utilizar los componentes de la vista en otras vistas como referencia. En la MVC, la predicción entre vistas se realiza como si el componente de la vista en otra vista fuera una referencia de interpredicción.

55 **[0067]** Como se muestra en la FIG. 3, un componente de la vista puede utilizar los componentes de la vista en otras vistas como referencia. Esto se llama predicción entre vistas. En la MVC, la predicción entre vistas se realiza como si el componente de la vista en otra visualización fuera una referencia de interpredicción. Las posibles referencias entre vistas se señalizan en la extensión de la MVC del conjunto de parámetros de secuencia (SPS)

65

(como se muestra en la Tabla 1 a continuación) y pueden ser modificadas por el proceso de construcción de la lista de imágenes de referencia, que habilita el ordenamiento flexible de las referencias de interpretación o de predicción entre vistas.

5

Tabla 1 - Tabla de sintaxis de extensión SPS MVC

	Descriptor
seq_parameter_set_mvc_extension() {	
num_views_minus1	ue(v)
for(i = 0; i <= num_views_minus1; i++)	
view_id[i]	ue(v)
for(i = 1; i <= num_views_minus1; i++) {	
num_anchor_refs_I0[i]	ue(v)
for(j = 0; j < num_anchor_refs_I0[i]; j++)	
anchor_ref_I0[i][j]	ue(v)
num_anchor_refs_I1[i]	ue(v)
for(j = 0; j < num_anchor_refs_I1[i]; j++)	
anchor_ref_I1[i][j]	ue(v)
}	
for(i = 1; i <= num_views_minus1; i++) {	
num_non_anchor_refs_I0[i]	ue(v)
for(j = 0; j < num_non_anchor_refs_I0[i]; j++)	
non_anchor_ref_I0[i][j]	ue(v)
num_non_anchor_refs_I1[i]	ue(v)
for(j = 0; j < num_non_anchor_refs_I1[i]; j++)	
non_anchor_ref_I1[i][j]	ue(v)
}	
num_level_values_signalled_minus1	ue(v)
for(i = 0; i <= num_level_values_signalled_minus1; i++)	
{	
level_idc[i]	u(8)
num_applicable_ops_minus1[i]	ue(v)
for(j = 0; j <= num_applicable_ops_minus1[i]; j++) {	
applicable_op_temporal_id[i][j]	u(3)
applicable_op_num_target_views_minus1[i][j]	ue(v)
for(k = 0; k <= applicable_op_num_target_views_minus1[i][j]; k++)	
applicable_op_target_view_id[i][j][k]	ue(v)
applicable_op_num_views_minus1[i][j]	ue(v)
}	
}	
}	
}	

10 **[0068]** En la extensión SPS MVC, para cada vista, se señala el número de vistas que se pueden utilizar para formar la lista de imágenes de referencia 0 y la lista de imágenes de referencia 1. Las relaciones de predicción para una imagen con anclaje, tal como se señala en la extensión SPS MVC, puede ser diferente de la relación de predicción para una imagen sin anclaje (señalizada en la extensión SPS MVC) de la misma vista.

15 **[0069]** La siguiente sección analizará la codificación de vídeo en 3D y multivista con respecto a la HEVC. En particular, las técnicas de ejemplo de esta divulgación son aplicables cuando se codifican dos o más vistas. La pluralidad de imágenes de vídeo para cada vista pueden denominarse componentes de textura de la vista. En 3D-HEVC, cada componente de textura de la vista puede tener un componente de profundidad de la vista

correspondiente. La MV-HEVC no utiliza componentes de profundidad de la vista. Los componentes de textura de la vista incluyen contenido de vídeo (por ejemplo, componentes de luma y croma de los valores de píxeles), y los componentes de profundidad de la vista (por ejemplo, para 3D-HEVC) pueden indicar profundidades relativas de los píxeles en los componentes de textura de la vista.

[0070] Las técnicas de ejemplo de esta divulgación se refieren a la codificación de datos de vídeo en 3D mediante la codificación de datos de textura o datos de textura y profundidad. En general, el término "textura" se usa para describir valores de luminancia (es decir, brillo o "luma") de una imagen y valores de crominancia (es decir, color o "croma") de la imagen. En algunos ejemplos, una imagen de textura puede incluir un conjunto de datos de luminancia y dos conjuntos de datos de crominancia, para matices azules (Cb) y matices rojos (Cr). En ciertos formatos de croma, tales como 4:2:2 o 4:2:0, los datos de croma se someten a un muestreo de frecuencia reducida con respecto a los datos de luma. Es decir, la resolución espacial de los componentes de crominancia es menor que la resolución espacial de los componentes de luminancia correspondientes, por ejemplo, un medio o un cuarto de la resolución de luminancia.

[0071] Los datos de profundidad en general describen valores de profundidad para los datos de textura correspondientes. Por ejemplo, una imagen de profundidad puede incluir un conjunto de píxeles de profundidad, cada uno de los cuales describe la profundidad para los datos de textura correspondientes. Los datos de profundidad pueden usarse para determinar la disparidad horizontal para los datos de textura correspondientes. Por lo tanto, un dispositivo que recibe los datos de textura y profundidad puede visualizar una primera imagen de textura para una vista (por ejemplo, una vista de ojo izquierdo) y usar los datos de profundidad para modificar la primera imagen de textura, para generar una segunda imagen de textura para la otra vista (por ejemplo, una vista de ojo derecho), desplazando valores de píxeles de la primera imagen en los valores de disparidad horizontal, determinados en base a los valores de profundidad. En general, la disparidad horizontal (o simplemente "disparidad") describe el desplazamiento espacial horizontal de un píxel en una primera vista hacia un píxel correspondiente de una segunda vista, donde los dos píxeles corresponden a la misma porción del mismo objeto tal como se representa en las dos vistas.

[0072] En otros ejemplos adicionales, pueden definirse datos de profundidad para los píxeles en una dimensión z perpendicular al plano de imagen, de manera que una profundidad asociada a un píxel dado se defina en relación con un plano de disparidad cero, definido para la imagen. Dicha profundidad puede usarse para crear disparidad horizontal para visualizar el píxel, de manera que el píxel se visualice de manera diferente para los ojos izquierdo y derecho, dependiendo del valor de profundidad de la dimensión z del píxel en relación con el plano de disparidad cero. El plano de disparidad cero puede cambiar para diferentes porciones de una secuencia de vídeo, y la cantidad de profundidad relativa al plano de disparidad cero también puede cambiar. Los píxeles situados en el plano de disparidad cero pueden definirse de forma similar para los ojos izquierdo y derecho. Los píxeles situados delante del plano de disparidad cero pueden visualizarse en diferentes ubicaciones para el ojo izquierdo y el derecho (por ejemplo, con disparidad horizontal) a fin de crear la percepción de que el píxel parece que salga de la imagen en la dirección z perpendicular al plano de imagen. Los píxeles situados tras el plano de disparidad nula pueden visualizarse ligeramente borrosos, para presentar una ligera percepción de profundidad, o pueden visualizarse en ubicaciones diferentes para el ojo izquierdo y derecho (por ejemplo, sin disparidad horizontal que esté opuesta a la de los píxeles situados delante del plano de disparidad nula). También pueden usarse muchas otras técnicas para transmitir o definir datos de profundidad para una imagen.

[0073] Para cada píxel en el componente de profundidad de la vista, puede haber uno o más píxeles correspondientes en el componente de textura de la vista. Por ejemplo, si las resoluciones espaciales del componente de profundidad de la vista y el componente de textura de la vista son iguales, cada píxel del componente de profundidad de la vista corresponde a un píxel del componente de textura de la vista. Si la resolución espacial del componente de profundidad de la vista es menor que la del componente de textura de la vista, entonces cada píxel en el componente de profundidad de la vista corresponde a múltiples píxeles en el componente de textura de la vista. El valor del píxel del componente de profundidad de la vista puede indicar la profundidad relativa del uno o más píxeles correspondientes en la vista de textura.

[0074] En algunos ejemplos, un codificador de vídeo señala datos de vídeo para los componentes de textura de la vista y los componentes de profundidad de la vista correspondientes para cada una de las vistas. Un decodificador de vídeo utiliza tanto los datos de vídeo de los componentes de textura de la vista como los componentes de profundidad de la vista para decodificar el contenido de vídeo de las vistas para su visualización. Una pantalla visualiza a continuación el vídeo multivista para producir vídeo en 3D.

[0075] El JCT-VC también está desarrollando extensiones ajustables a escala de la HEVC. La FIG. 4 es un diagrama conceptual que ilustra un ejemplo de codificación de vídeo ajustable a escala. Si bien la FIG. 4 se describe con respecto a las normas H.264/AVC y SVC, debería entenderse que las capas similares se pueden codificar usando otros esquemas de codificación de vídeo multicapa, incluida la extensión ajustable a escala de la HSVC. El ejemplo de la FIG. 4 muestra capas codificadas usando el mismo códec. En otros ejemplos, las capas pueden codificarse usando un códec de múltiples normas. Por ejemplo, una capa base puede codificarse usando H.264/AVC, mientras que una capa de mejora puede codificarse usando una extensión ajustable a escala a la

HEVC. Por lo tanto, las referencias a la SVC a continuación pueden aplicarse a la codificación de vídeo ajustable a escala en general, y no están restringidas a la extensión ajustable a escala de la H.264/SVC.

[0076] En SVC, las ajustabilidades a escala se pueden habilitar en tres dimensiones que incluyen, por ejemplo, la espacial, la temporal y la cualitativa (representadas como unas dimensiones de tasa de bits o relación de señal-ruido (SNR)). En general, normalmente se puede lograr una mejor representación añadiendo a una representación en cualquier dimensión. Por ejemplo, en el ejemplo de la FIG. 4, la capa 0 se codifica en un Cuarto Formato Intermedio Común (QCIF) que tiene una frecuencia de trama de 7,5 Hz y una tasa de bits de 64 kilooctetos por segundo (KBPS). Además, la capa 1 se codifica en QCIF con una frecuencia de trama de 15 Hz y una tasa de bits de 64 KBPS, la capa 2 se codifica en CIF con una frecuencia de trama de 15 Hz y una tasa de bits de 256 KBPS, la capa 3 se codifica en QCIF con una frecuencia de trama de 7,5 Hz y una tasa de bits de 512 KBPS y la capa 4 se codifica en 4CIF con una frecuencia de trama de 30 Hz y una tasa de bits de megabytes por segundo (MBPS). Debería entenderse que el número, el contenido y la disposición particulares de las capas que se muestran en la FIG. 4 se proporcionan solamente con fines de ejemplo.

[0077] En cualquier caso, una vez que un codificador de vídeo (tal como el codificador de vídeo 20) ha codificado el contenido de la dicha manera ajustable a escala, un descodificador de vídeo (tal como el descodificador de vídeo 30) puede usar una herramienta extractora para adaptar el contenido actual suministrado de acuerdo con los requisitos de la aplicación, que pueden depender, por ejemplo, del cliente o del canal de transmisión.

[0078] En la SVC, las imágenes que tienen la capa espacial y cualitativa más baja son típicamente compatibles con la norma H.264/AVC. En el ejemplo de la FIG. 4, las imágenes con la capa espacial y cualitativa más bajas (imágenes en la capa 0 y capa 1, con resolución de QCIF) pueden ser compatibles con la norma H.264/AVC. Entre ellas, aquellas imágenes del nivel temporal más bajo forman la capa base temporal (por ejemplo, capa 0). Esta capa base temporal (por ejemplo, capa 0) se puede mejorar con imágenes de niveles temporales superiores (por ejemplo, capa 1).

[0079] Además de la capa compatible con la norma H.264/AVC, se pueden añadir varias capas de mejora espacial y/o cualitativas para proporcionar ajustabilidades a escala, espaciales y/o cualitativas. Cada capa de mejora espacial o cualitativa en sí misma puede ser temporalmente ajustable a escala, con la misma estructura de ajustabilidad a escala temporal que la capa compatible con la norma H.264/AVC.

[0080] Como se ha descrito anteriormente, el término "capa" se usa en general en esta divulgación para referirse a capas base individuales o capas de mejora individuales en el sentido de la codificación de vídeo ajustable a escala, o vistas de textura y/o vistas de profundidad en el sentido de 3D y codificación de vídeo multivista. Una vez que las imágenes para cada una de las capas (por ejemplo, capas MV-HEVC o capas SHVC) se descodifican, por ejemplo, mediante el descodificador de vídeo 30 o el bucle de reconstrucción del codificador de vídeo 20, las imágenes para la capa descodificada se almacenan en un DPB. Un DPB es un búfer o memoria que se usa para almacenar imágenes, y dentro de esta divulgación, imágenes de capas descodificadas cuando se utilizan técnicas de codificación de vídeo multicapa. Las imágenes de las capas descodificadas almacenadas en el DPB pueden usarse como referencias para la interpredicción (incluida la predicción por compensación de movimiento, entre vistas y entre capas), para el reordenamiento de salida y el retardo de salida.

[0081] De acuerdo con diversos ejemplos de esta divulgación, un DPB puede dividirse en subunidades (por ejemplo, llamadas sub-DPB). Cada sub-DPB puede configurarse para almacenar una o más imágenes, o más en general, información de imagen, para un tipo de capa particular. Por ejemplo, un DPB puede dividirse de manera que un sub-DPB esté configurado para almacenar imágenes de una capa base para un proceso de codificación de vídeo ajustable a escala. Se puede configurar otro sub-DPB para almacenar imágenes de una primera capa de mejora de un proceso de codificación de vídeo ajustable a escala. Aún se puede configurar otro sub-DPB para almacenar imágenes de una segunda capa de mejora de un proceso de codificación de vídeo ajustable a escala, y así sucesivamente.

[0082] De acuerdo con diversos ejemplos de esta divulgación, como se describirá con más detalle a continuación, el codificador de vídeo 20 también se puede configurar para gestionar cada sub-DPB independientes entre sí, como por ejemplo, en términos de marcado de imágenes para su eliminación del DPB, eliminación de la imagen descodificada del DPB, así como la asignación de tamaño para cada sub-DPB. Sin embargo, el codificador de vídeo 20 puede vincular la salida de imágenes descodificadas de diferentes sub-DPB, por ejemplo, en base a los tiempos de salida respectivos y/o los valores de recuento de orden de imágenes (POC) asociados con cada imagen descodificada.

[0083] Uno o ambos del codificador de vídeo 20 y descodificador de vídeo 30 (y/o uno o más componentes del mismo) pueden implementar múltiples modos diferentes de operaciones basadas en sub-DPB. Por ejemplo, dos modos de operación basados en sub-DPB pueden diferenciarse en términos de la forma en que las imágenes están asociadas con cada sub-DPB. De acuerdo con el primer modo (denominado en el presente documento "modo específico de capa"), cada capa en el punto de operación puede asociarse con un sub-DPB separado, y el número de sub-DPB puede ser igual al número de capas en el conjunto de capas correspondiente al punto de

operación. De acuerdo con un segundo modo (denominado en el presente documento "modo específico de resolución"), todas las imágenes que comparten la misma resolución, formato de croma y profundidad de bits comparten el mismo sub-DPB. El codificador de vídeo 20 puede especificar el modo específico de capa con relativa facilidad. Sin embargo, el modo específico de capa puede ser menos eficaz en términos de utilización de memoria DPB, en comparación con el modo específico de resolución.

[0084] La FIG. 5 es un diagrama conceptual que muestra un DPB 900 que está configurado para almacenar diferentes componentes de capa descodificados (por ejemplo, imágenes para diferentes tipos de capas) en diferentes subunidades. DPB 900 puede formar parte de un descodificador de vídeo y/o codificador de vídeo (por ejemplo, codificador de vídeo 20 y descodificador de vídeo 30). Cada uno de los sub-DPB 910A-910D representa ubicaciones de almacenamiento para imágenes de diferentes tipos de capas descodificadas. Por ejemplo, el sub-DPB 910A puede configurarse para almacenar una o más imágenes de textura de una vista base para un flujo de bits de vídeo compatible con MV-HEVC, mientras que el sub-DPB 910B puede usarse para almacenar imágenes de textura de una vista dependiente para el flujo de bits de vídeo compatible con MV-HEVC. De manera similar, el sub-DPB 910C puede usarse para almacenar imágenes de una capa base en un proceso de codificación de vídeo ajustable a escala (por ejemplo, SHVC), mientras que el sub-DPB 910D puede usarse para almacenar un nivel de imágenes para una capa de mejora en el proceso de codificación de vídeo ajustable a escala. Se pueden configurar sub-DPB adicionales para almacenar imágenes de niveles adicionales de la capa de mejora o imágenes de vistas adicionales dependientes de la textura. Cada uno de los sub-DPB 910A-D contiene unidades separadas que están configuradas para almacenar una o más imágenes de una capa descodificada para el tipo de capa particular asociado con el sub-DPB. Cabe destacar que los datos de vídeo codificados con MV-HEVC y los datos de vídeo codificados con SHVC no se descodifican necesariamente al mismo tiempo. Es decir, los datos de vídeo codificados SHVC y MV-HEVC no suelen estar en el mismo flujo de bits de vídeo codificado, sino que típicamente están codificados por separado.

[0085] El descodificador de vídeo 30 puede configurarse además para eliminar imágenes de capas descodificadas de la pluralidad de los sub-DPB 910A-D. En algunos ejemplos, la eliminación de imágenes de cada sub-DPB (por ejemplo, sub-DPB 910A) puede gestionarse por separado (es decir, las imágenes pueden eliminarse de un sub-DPB independientemente de la eliminación de otras imágenes de otros sub-DPB). En otros ejemplos, las imágenes pueden eliminarse de uno o más sub-DPB en base a un punto de operación. Un punto de operación particular está asociado con un número particular de capas que se descodifican para que se genere una imagen particular. Por ejemplo, para SHVC, un punto de operación solo puede requerir una imagen de una capa base a descodificar. En otro punto de operación para SHVC, una imagen de una capa base y las imágenes de una o más capas de mejora pueden descodificarse para producir una imagen de salida. En el punto de operación que requiere tanto una imagen de una capa base como imágenes de una o más capas de mejora, la eliminación de imágenes se puede realizar para cada sub-DPB utilizado para ese punto de operación (es decir, cada sub-DPB que almacena imágenes para una capa que se usa para la imagen a generar). Es decir, una imagen de una capa base y las imágenes correspondientes de las una o más capas de mejora asociadas con un punto de operación dado pueden eliminarse de sus respectivos sub-DPB al mismo tiempo.

[0086] El codificador de vídeo 20 y el descodificador de vídeo 30 pueden configurarse adicionalmente para marcar componentes de capa descodificados (por ejemplo, imágenes de capas) como no utilizadas de referencia o utilizadas de referencia en la pluralidad de los sub-DPB 910A-D. Nuevamente, el proceso de marcado de imágenes como no utilizadas de referencia puede realizarse por separado para cada sub-DPB. Es decir, el codificador de vídeo 20 y el descodificador de vídeo 30 pueden configurarse para realizar un proceso de marcado en las imágenes en cada una de la pluralidad de sub-DPB de forma independiente, en el que el proceso de marcado marca las imágenes como utilizadas de referencia o las imágenes como no utilizadas de referencia. En otro ejemplo, el proceso de marcado puede realizarse para todos los sub-DPB que almacenan imágenes para un número particular de capas relacionadas con un punto de operación particular.

[0087] En HEVC y otras normas de codificación de vídeo, la operación del DPB 900 a menudo se especifica en relación con un descodificador de referencia hipotético (HRD). El codificador de vídeo 20 y el descodificador de vídeo 30 pueden configurarse para gestionar el DPB 900 para realizar diversas acciones, que incluyen el marcado de imágenes descodificadas almacenadas en el DPB 900 como "no utilizadas de referencia" (es decir, no pueden usarse como imagen de referencia para un proceso de interpredicción), marcado de una imagen descodificada para su salida (por ejemplo, a un dispositivo de visualización 32) y marcado de una imagen descodificada para su eliminación del DPB 900 (también conocido como "expulsión"). En HEVC, el proceso de expulsión incluye tanto el marcado de una imagen para su salida como el marcado de una imagen para su eliminación. Una imagen típicamente se elimina (es decir, se echa fuera) del DPB 900 cuando la imagen ya no se necesita como imagen de referencia para la interpredicción y ya no se necesita para su salida.

[0088] La conformidad de los flujos de bits se especifica definiendo el comportamiento del HRD. El DPB 900 puede considerarse un componente del DRH. Como se ha descrito anteriormente, el DPB 900 puede implementarse como una o más unidades de memoria configuradas para almacenar imágenes reconstruidas en un codificador de vídeo (por ejemplo, codificador de vídeo 20) o imágenes descodificadas en un descodificador de vídeo (por ejemplo, descodificador de vídeo 30). En algunas propuestas para SHVC y MV-HEVC, las operaciones

DPB no se especifican claramente, pero una suposición simple es que todas las imágenes con el mismo tiempo de salida llegan al DPB 900 (es decir, se almacenan en el DPB 900) en el mismo instante de tiempo. La salida de imágenes del DPB 900 se determina en base a cómo se definan los parámetros HRD.

5 **[0089]** Por ejemplo, los parámetros HRD pueden definirse para la conformidad del orden de salida. A modo de ejemplo, el orden de salida de las imágenes de capa descodificadas puede especificarse mediante uno o más elementos sintácticos cuyo valor especifica el número de imágenes presentes en una capa particular. Para mencionar otro ejemplo, en la conformidad del tiempo de salida, el tiempo de salida puede especificarse mediante uno o más elementos sintácticos cuyos valores indican tiempos de salida de la imagen. El descodificador de vídeo
10 30 puede configurarse para derivar ajustes de HRD a partir de los valores de los parámetros señalizados en un flujo de bits de vídeo codificado. Los parámetros de HRD pueden señalizarse en una estructura de formato de datos (por ejemplo, llamada `hrd_parameters()`). Los parámetros de ejemplo pueden incluir un mensaje de información de mejora complementaria (SEI) del período de almacenamiento en búfer y un mensaje SEI de temporización de imagen. En algunos ejemplos, también se puede definir un mensaje SEI de información de la unidad de descodificación (DU). En algunos ejemplos, la salida de las imágenes en DPB 900 se produce de manera que todas las imágenes asociadas con el mismo tiempo de salida se generan en el mismo instante. En ejemplos de esta divulgación, las imágenes en diferentes sub-DPB pueden generarse al mismo tiempo en el caso de que cada una de esas imágenes (es decir, las imágenes en los diferentes sub-DPB) estén asociadas con el mismo tiempo de salida.

20 **[0090]** En algunas propuestas de ejemplo para HEVC, los parámetros relacionados con las operaciones DPB se señalizan en uno o más de un conjunto de parámetros de secuencia (SPS) y/o un conjunto de parámetros de vídeo (VPS). Un SPS es una estructura sintáctica que contiene elementos sintácticos que se aplican a cero o más secuencias de vídeo codificadas (CVS) enteras, como se indica con los valores de los elementos sintácticos que se encuentran en un conjunto de parámetros de imagen (PPS) y/o una cabecera de fragmento. Un VPS es una estructura sintáctica que contiene un elemento sintáctico que se aplica a cero o más CVS completos según se determina con el valor del elemento sintáctico encontrado en una cabecera de SPS, PPS y/o fragmento. En general, los elementos sintácticos señalizados en un VPS se aplicarán a un mayor número de imágenes que el número de imágenes a las que se aplica el elemento sintáctico señalado en un SPS. Como se define en HEVC,
25 un CVS es una secuencia de unidades de acceso que consiste, en el orden de descodificación, de una unidad de acceso (AU) que contiene una imagen de acceso aleatorio intra (IRAP) asociada con un `nuh_layer_id` igual a un valor de 0 y con un `NoRasOutputFlag` igual a un valor de 1 seguido de cero o más AU adicionales hasta la próxima AU (es decir, hasta, pero sin incluir la siguiente AU), en el orden de descodificación, que contiene una imagen IRAP con id de capa `nuh` igual a un valor de 0 y `NoRasOutputFlag` igual a un valor de 1. Una unidad de acceso es un conjunto de unidades de la capa de abstracción de red (NAL) que son consecutivas en el orden de descodificación y contienen una o más imágenes codificadas que cada una tiene el mismo tiempo de salida.

35 **[0091]** Los elementos sintácticos de ejemplo que están relacionados con las operaciones DPB incluyen, pero no se limitan a, el elemento sintáctico `max_dec_pic_buffering_minus1`, el elemento sintáctico `max_num_reorder_pics` y el elemento sintáctico `max_latency_increase_plus1`, que se pueden señalar en el SPS y/o en el VPS. El elemento sintáctico `max_dec_pic_buffering_minus1`, el elemento sintáctico `max_num_reorder_pics` y el elemento sintáctico `max_latency_increase_plus1` pueden denominarse colectivamente parámetros de DPB y/o parámetros de sub-DPB.

45 **[0092]** El valor del elemento sintáctico `max_dec_pic_buffering_minus1` especifica el tamaño máximo permitido de un DPB o sub-DPB para un CVS en unidades de búferes de imágenes. Es decir, un DPB o un sub-DPB pueden dividirse en uno o más búferes de imágenes, donde cada búfer de imagen almacena una imagen (por ejemplo, una imagen asociada con una capa particular). En algunos ejemplos, el valor de `max_dec_pic_buffering_minus1` está limitado a un intervalo de 0 a un tamaño máximo predeterminado de DPB y/o sub-DPB.

50 **[0093]** El valor del elemento sintáctico `max_num_reorder_pics` especifica el número máximo permitido de imágenes que pueden almacenarse en un DPB y/o sub-DPB y que pueden preceder a cualquier imagen en el CVS en el orden de descodificación e ir detrás de esa misma imagen en el orden de salida. El valor del elemento sintáctico `max_latency_increase_plus1`, cuando no es igual a cero (es decir, en base al valor del elemento sintáctico `max_latency_increase_plus1` es un valor distinto de cero), es una indicación del tiempo de latencia que se utiliza para determinar el número máximo de imágenes que puede almacenarse en un DPB y/o sub-DPB que puede preceder a cualquier imagen en el CVS en el orden de salida e ir detrás de esa misma imagen en el orden de descodificación. Si el número de imágenes en un DPB y/o sub-DPB sobrepasa el número máximo de imágenes indicado por el valor del elemento sintáctico `max_num_reorder_pics` o el valor determinado a partir del valor del elemento sintáctico `max_latency_increase_plus1`, se deben eliminar una o más imágenes del DPB y/o el sub-DPB.

60 **[0094]** En un ejemplo, el codificador de vídeo 20 puede configurarse para generar y señalar los parámetros de DPB y/o parámetros de sub-DPB antes mencionados para cada subcapa, por ejemplo, si (es decir, en base a) el valor de un elemento sintáctico de compuerta (por ejemplo, `sub_layer_ordering_info_present_flag`) es igual a 1. Como se ha descrito anteriormente, una subcapa es una capa temporal ajustable a escala de un flujo de bits temporal ajustable a escala. De acuerdo con este ejemplo, cuando la señalización de la subcapa no está permitida,
65

solo se señala un valor para cada uno de los elementos sintácticos, y los valores señalizados de cada uno de estos elementos sintácticos de parámetro sub-DPB (es decir, el elemento sintáctico max_dec_pic_buffering_minus1, el elemento sintáctico max_num_reorder_pics y el elemento sintáctico max_latency_increase_plus1) se infieren (es decir, se determinan sin la recepción de señalización explícita) para que sean iguales en todas las subcapas temporales. En caso contrario, cuando se permite la señalización de la subcapa (por ejemplo, el valor del elemento sintáctico sub_layer_ordering_info_present_flag es igual a 0), los valores de los elementos sintácticos del parámetro sub-DPB se señalizan explícitamente para cada subcapa. La señalización de los parámetros de sub-DPB en el SPS se destaca en la Tabla 2 a continuación en cursiva. Una señalización similar se puede utilizar en el VPS.

Tabla 2 - Parámetros de DPB en el SPS

	Descriptor
seq_parameter_set_rbsp() {	
sps_video_parameter_set_id	u(4)
if(nuh_layer_id == 0) {	
sps_max_sub_layers_minus1	u(3)
sps_temporal_id_nesting_flag	u(1)
profile_tier_level(1, sps_max_sub_layers_minus1)	
}	
sps_seq_parameter_set_id	ue(v)
chroma_format_idc	ue(v)
if(chroma_format_idc == 3)	
separate_colour_plane_flag	u(1)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
conformance_window_flag	u(1)
if(conformance_window_flag) {	
conf_win_left_offset	ue(v)
conf_win_right_offset	ue(v)
conf_win_top_offset	ue(v)
conf_win_bottom_offset	ue(v)
}	
bit_depth_luma_minus8	ue(v)
bit_depth_chroma_minus8	ue(v)
log2_max_pic_order_cnt_lsb_minus4	ue(v)
sps_sub_layer_ordering_info_present_flag	u(1)
<i>for(i = (sps_sub_layer_ordering_info_present_flag ? 0 : sps_max_sub_layers_minus1); i <= sps_max_sub_layers_minus1; i++) {</i>	
sps_max_dec_pic_buffering_minus1[i]	ue(v)
sps_max_num_reorder_pics[i]	ue(v)
sps_max_latency_increase_plus1[i]	ue(v)
<i>}</i>	
...	

[0095] En la versión base de la HEVC (a veces llamada HEVC 1), solo está presente una sola capa (por ejemplo, como se indica con el elemento sintáctico id de capa nuh que tiene un valor igual a 0) y, por lo tanto, el conjunto de capas decodificado y el conjunto de capas generado son el mismo. En el contexto de la codificación de vídeo multicapa (por ejemplo, SHVC o MV-HEVC), el número de conjuntos de capas y conjuntos de capas de salida no es necesariamente el mismo, ya que el número total de conjuntos de capas no se puede generar para cada punto de operación. Es decir, los puntos de operación SHVC a menudo incluyen la decodificación de una capa base y una o más capas de mejora. Del mismo modo, los puntos de operación MV-HEVC a menudo incluyen la

descodificación de dos o más vistas. Sin embargo, no todas las capas o vistas se pueden usar para generar una imagen dada.

[0096] Se analizarán ahora las técnicas para eliminar imágenes del DPB. Ejemplos de técnicas para HEVC implican dos maneras de especificar la eliminación de imágenes del DPB. Incluyen un proceso de eliminación de DPB basado en imágenes y un proceso de eliminación de DPB basado en AU. En un proceso de eliminación de DPB basado en imágenes, las imágenes en una AU pueden eliminarse individualmente en diferentes momentos. En las operaciones de eliminación de DPB basadas en AU, todas las imágenes en la AU se eliminan al mismo tiempo. Las propuestas actuales para las operaciones de eliminación de DPB para SHVC y MV-HEVC no especifican claramente cómo se deben eliminar las imágenes.

[0097] Las propuestas anteriores para señalar los parámetros de DPB, incluidos los parámetros de operaciones sub-DPB (es decir, parámetros de sub-DPB), como los propuestos en Ramasubramonian, y col., "MV-HEVC/SHVC HLS: On DPB Operations", Equipo de colaboración conjunta sobre extensiones de codificación de vídeo en 3D de ITU-T SG 16 WP 3 e ISO/IEC JTC 1/SC 29/WG 11, 5ª reunión: Viena, AT, 27 de julio - 2 de agosto de 2013 (en lo sucesivo, "JCTVC-N0093_v4") pueden ser subóptimas. A modo de ejemplo, de acuerdo con el esquema de señalización de operaciones basadas en sub-DPB en JCTVC-N0093_v4, los tres parámetros relacionados con DPB y sub-DPB (es decir, el elemento sintáctico `max_dec_pic_buffering_minus1`, el elemento sintáctico `max_num_reorder_pics` y el elemento sintáctico `max_latency_increase_plus1` descritos anteriormente) se señalizan para cada conjunto de capas de salida, cada capa en el conjunto de capas de salida y para cada subcapa temporal dentro de cada capa. Un esquema de señalización de este tipo puede ser subóptimo.

[0098] En particular, los elementos sintácticos de señalización que indican el número máximo de imágenes reordenadas y la latencia máxima para cada capa en cada conjunto de capas de salida pueden ser redundantes, lo que puede provocar potencialmente el desperdicio de recursos informáticos (por ejemplo, recursos de procesador, capacidad de almacenamiento, etc.) y/o ancho de banda de la red. Por ejemplo, la salida de una o más imágenes de una capa de salida establecida en una unidad de acceso (AU) puede ser simultánea. Además, la salida de imágenes desde una segunda AU antes de que se generen todas las imágenes en una primera AU puede no estar permitida. Por lo tanto, la señalización de los parámetros de sub-DPB para cada capa en un conjunto de capas de salida puede ser redundante, ya que los diferentes valores para los parámetros de sub-DPB pueden no ser posibles. Como tal, puede ser redundante enviar información de reordenamiento y/o latencia para cada capa en un conjunto de capas de salida.

[0099] Para mitigar o eliminar las ineficacias y otros problemas potenciales causados por dicha redundancia de datos señalizados, el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden implementar una o más de las técnicas descritas en el presente documento. En uno o más ejemplos de la divulgación, el codificador de vídeo 20 puede configurarse para señalar un conjunto de parámetros de sub-DPB para cada conjunto de capas de salida correspondiente a un punto de operación. El único conjunto de parámetros de sub-DPB señalado para cada conjunto de capas de salida se utiliza por el descodificador de vídeo 30 para cada capa respectiva en cada conjunto de capas de salida. De esta manera, se aumenta la eficacia de la señalización, ya que se evita la señalización redundante de los parámetros de sub-DPB. Además, los valores de los parámetros de sub-DPB pueden armonizarse en todas las capas de un conjunto de capas de salida.

[0100] En un ejemplo de la divulgación, para el conjunto de capas de salida de cada punto de operación, el codificador de vídeo 20 puede configurarse para generar y señalar, en el VPS, solo un valor (es decir, un valor único) de un elemento sintáctico indicativo del número máximo de imágenes reordenadas y solo un valor (es decir, un solo valor) de un elemento sintáctico indicativo de la latencia máxima. Es decir, para cada punto de operación relacionado con un VPS particular, el codificador de vídeo 20 genera un elemento sintáctico del número máximo de imágenes reordenadas y un elemento sintáctico de la latencia máxima. Por consiguiente, el número total de elementos sintácticos de imágenes de reordenamiento y elementos sintácticos de latencia máxima en cada VPS depende del número de puntos de operación definidos para el flujo de bits de vídeo. Como tal, se señalizan menos instancias de elementos sintácticos de imágenes de reordenamiento y elementos sintácticos de latencia máxima en comparación con las propuestas anteriores a la HEVC donde los elementos sintácticos de imágenes de reordenamiento y los elementos sintácticos de latencia máxima se señalizan para cada conjunto de capas de salida, cada capa en el conjunto de capas de salida, y para cada subcapa temporal dentro de cada capa.

[0101] Como se describirá con más detalle a continuación con referencia a la implementación del primer ejemplo, el elemento sintáctico indicativo del número máximo de imágenes reordenadas y el elemento sintáctico indicativo de la latencia máxima se pueden señalar en una tabla de tamaños de DPB y se pueden transmitir en un VPS. En otros ejemplos de la divulgación, el elemento sintáctico indicativo del número máximo de imágenes reordenadas y el elemento sintáctico indicativo de la latencia máxima pueden transmitirse en otras estructuras de datos que incluyen un PPS.

[0102] Al recibir el elemento sintáctico indicativo del número máximo de imágenes reordenadas y el elemento sintáctico indicativo de la latencia máxima, el descodificador de vídeo 30 puede aplicar los valores respectivos de

los elementos sintácticos indicativos del número máximo de imágenes reordenadas e indicativos de la latencia máxima a la capa de salida establecida para un punto de operación particular.

5 **[0103]** En un ejemplo, los elementos sintácticos indicativos del número máximo de imágenes reordenadas e
 10 indicativos de la latencia máxima pueden corresponder a solo una capa de una capa de salida. Por ejemplo, los
 elementos sintácticos pueden corresponder a la capa que tiene una identificación de capa más alta (por ejemplo,
 como se indica con el elemento sintáctico `nuh_layer_id`). En este ejemplo, el codificador de vídeo 20 puede
 configurarse para señalar un indicador para indicar al descodificador de vídeo 30 si el valor del número máximo
 de imágenes reordenadas y el valor de la latencia máxima están asociados con la capa con el valor más grande
 de `nuh_layer_id` en el punto de operación. El descodificador de vídeo 30 se configuraría para inferir el valor de los
 elementos sintácticos para que el conjunto de capas de salida sea el mismo que el valor recibido para la capa que
 tiene el `id` de capa más alto.

15 **[0104]** En otro ejemplo de la divulgación, el codificador de vídeo 20 puede señalar un valor de identificación
 (por ejemplo, usando el elemento sintáctico `nuh_layer_id`) para indicar al descodificador de vídeo 30 la capa con
 la cual el valor del número máximo de imágenes reordenadas y el valor de la latencia máxima está asociada.
 Nuevamente, el descodificador de vídeo 30 se configuraría para inferir el valor de los elementos sintácticos para
 que el conjunto de capas de salida sea el mismo que el valor recibido para la capa que tiene el `id` de capa más
 alto.

20 **[0105]** En otro ejemplo de la divulgación, para cada punto de operación, el codificador de vídeo 20 puede
 configurarse para señalar un indicador para indicar al descodificador de vídeo 30 si los parámetros de sub-DPB
 (por ejemplo, un elemento sintáctico que indica el número máximo de imágenes reordenadas y/o un elemento
 sintáctico que indica la latencia máxima) solo se señala para la capa con el valor más grande de `nuh_layer_id` en
 25 el conjunto de capas de salida (es decir, la capa más alta), o que los parámetros de sub-DPB se señalizan para
 cada capa en el conjunto de capas de salida. Si el valor del indicador indica (es decir, en base a lo que el indicador
 indica) que los parámetros de sub-DPB solo se señalizan para la capa más alta, el descodificador de vídeo 30
 puede configurarse para inferir el valor de los parámetros de sub-DPB para la capa de salida establecida para
 tener el mismo valor que los parámetros de sub-DPB para la capa más alta.

30 **[0106]** En otro ejemplo de la divulgación, para cada conjunto de capas de salida, el codificador de vídeo 20 puede
 señalar un indicador para especificar al descodificador de vídeo 30 si los parámetros de sub-DPB (por ejemplo,
 elementos sintácticos indicativos del tamaño de sub-DPB, el número máximo de imágenes reordenadas, y/o la
 latencia máxima) se señalizan solo para una subcapa (por ejemplo, la subcapa con ID temporal igual a 0) o para
 35 más de una subcapa. Por ejemplo, el codificador de vídeo 20 puede generar el indicador a un primer valor para
 indicar que la información de los parámetros del sub-DPB se señalizan solo para una subcapa. En este caso, el
 descodificador de vídeo 30 inferirá entonces que el valor de los parámetros de sub-DPB para todas las otras
 subcapas tiene el mismo valor que los parámetros de sub-DPB realmente señalizados para una subcapa (por
 ejemplo, la subcapa con ID temporal igual a 0).

40 **[0107]** A la inversa, el codificador de vídeo 20 puede establecer el indicador en un segundo valor para indicar
 que los parámetros de sub-DPB pueden señalizarse para múltiples subcapas. En un ejemplo, cuando el valor del
 indicador indica (es decir, en base a lo que el indicador indica) que los parámetros de sub-DPB pueden señalizarse
 para múltiples subcapas, el codificador de vídeo 20 puede configurarse para generar elementos sintácticos para
 45 los parámetros de sub-DPB de cada subcapa en un conjunto de capas de salida. En otro ejemplo, como se
 explicará con más detalle a continuación con referencia a las Tablas 3 y 4, el codificador de vídeo 20 puede
 configurarse para generar un segundo indicador que indique, para cada subcapa del conjunto de capas de salida,
 si o no se señalizan parámetros de sub-DPB adicionales para una subcapa particular. Cuando el segundo indicador
 indica que se deben señalar parámetros adicionales de sub-DPB para una subcapa particular, el codificador de
 50 vídeo 20 genera y señala los parámetros de sub-DPB adicionales para la subcapa particular. Si el segundo
 indicador indica que no se señalarán parámetros de sub-DPB adicionales para una subcapa particular, el
 descodificador de vídeo 30 puede configurarse para inferir los valores de los parámetros de sub-DPB para la
 subcapa particular a partir de los parámetros de sub-DPB previamente señalizados para el conjunto de capas de
 salida (por ejemplo, a partir de los parámetros de sub-DPB para la subcapa que tiene un ID temporal igual a 0, o
 55 a partir de los parámetros de sub-DPB para la subcapa anterior). En algunos ejemplos, cuando el segundo
 indicador indica que los parámetros de sub-DPB no están señalizados para una subcapa particular, los parámetros
 de sub-DPB de las subcapas posteriores a esa subcapa particular tampoco están señalizados, y el descodificador
 de vídeo 30 puede configurarse para inferir el valor de los parámetros de sub-DPB para esas subcapas posteriores
 a partir de los parámetros de sub-DPB previamente señalizados para el conjunto de capas de salida.

60 **[0108]** El descodificador de vídeo 30 puede configurarse adicionalmente para realizar un proceso de gestión de
 sub-DPB (por ejemplo, marcado de imágenes para su salida, marcado de imágenes no utilizadas de referencia,
 generación de imágenes y/o eliminación de imágenes) de acuerdo con las técnicas de señalización de parámetros
 de sub-DPB de esta divulgación. Por ejemplo, el descodificador de vídeo 30 puede configurarse para realizar el
 65 proceso de expulsión en un nivel de unidad de acceso, en lugar de un nivel de imagen. Esto se debe a que, de
 acuerdo con las técnicas de esta divulgación, los parámetros de sub-DPB se aplican a un conjunto completo de

capas de salida y no solo a un subconjunto de las capas. También en este sentido, se elimina la necesidad de verificar los valores de los parámetros de sub-DPB para los ID de capa particulares, ya que cada capa de un conjunto de capas de salida particular comparte los mismos parámetros de sub-DPB. Más detalles de los procesos de gestión de DPB de acuerdo con las técnicas de esta divulgación se describen a continuación con referencia a la implementación del primer ejemplo.

[0109] Las siguientes técnicas son ejemplos adicionales de técnicas de señalización para señalar los parámetros de sub-DPB. En un ejemplo, para cada punto de operación, el codificador de vídeo 20 puede señalar un indicador para especificar al descodificador de vídeo 30 si los parámetros de sub-DPB están señalizados. De forma alternativa, el descodificador de vídeo 30 puede inferir un valor del indicador, en los casos en que el codificador de vídeo 20 no señala el indicador. En otro ejemplo, para cada punto de operación, el codificador de vídeo 20 puede señalar un indicador para indicar al descodificador de vídeo 30 si: a) los tamaños de sub-DPB solo se señalizan para el sub-DPB del índice más alto y para los otros sub-DPB se infiere que es igual al tamaño del sub-DPB con el índice más alto, o b) los tamaños de sub-DPB se señalizan para cada sub-DPB en el punto de operación.

[0110] Las siguientes secciones describen implementaciones de ejemplo de las técnicas de esta divulgación con respecto a las técnicas de codificación de HEVC. Las implementaciones de ejemplo se describirán utilizando la terminología y las definiciones de las memorias descriptivas de HEVC, SHVC, MV-HEVC y/o 3D-HEVC. Las definiciones de cualquier término que no se defina explícitamente a continuación se pueden encontrar en las memorias descriptivas de HEVC, SHVC, MV-HEVC y/o 3D-HEVC, según se han referenciado anteriormente.

[0111] En un primer ejemplo de implementación de las técnicas descritas en el presente documento, se describen técnicas para señalar parámetros de DPB y sub-DPB. Ejemplos de elementos sintácticos asociados con el primer ejemplo de implementación se muestran en la Tabla 3 y la Tabla 4 a continuación. Los elementos sintácticos relacionados con las técnicas de esta descripción se distinguen en la Tabla 3 y la Tabla 4 en **negrita y subrayado**. Las supresiones a las contribuciones HEVC anteriores se muestran con a. Las porciones subrayadas de la Tabla 1 y la Tabla 2 están dirigidas a los cambios introducidos por la implementación de ejemplo #1 a la sintaxis existente.

Tabla 3 - Sintaxis de la extensión del conjunto de parámetros de vídeo

vps_extension() {	Descriptor
avc_base_layer_flag	u(1)
...	
cross_layer_irap_aligned_flag	u(1)
<u>dpb_size_table()</u>	
direct_dep_type_len_minus2	ue(v)
....	
}	

Como se muestra en la Tabla 3, el codificador de vídeo 20 puede configurarse para señalar una tabla de tamaños de DPB (dpb_size_table()) en una sintaxis de extensión de VPS. La tabla de tamaños de DPB puede incluir parámetros relacionados con DPB y sub-DPB, incluidos los elementos sintácticos que tienen valores indicativos del tamaño de sub-DPB, el número máximo de imágenes reordenadas y/o la latencia máxima (los parámetros de sub-DPB).

Tabla 4 - Sintaxis de la tabla de tamaños de DPB

dpb_size_table() {	
for(i = 1; i < NumOutputLayerSets; i++) {	
sub_layer_flag_info_present_flag[i]	u(1)
for(j = 0; j <= vps_max_sub_layers_minus1; j++) {	
if(j > 0 && sub_layer_flag_info_present_flag[i] &&	
sub_layer_dpb_info_present_flag[i][j - 1])	
<u>sub_layer_dpb_info_present_flag[i][j]</u>	u(1)
if(sub_layer_dpb_info_present_flag[i][j]) {	
for(k = 0; k < NumSubDpbs[i]; k++)	

<code>max_vps_dec_pic_buffering_minus1[i][k][j]</code>	ue(v)
<code>max_vps_num_reorder_pics[i][j]</code>	ue(v)
<code>max_vps_latency_increase_plus1[i][j]</code>	ue(v)
<code>}</code>	
<code>}</code>	
<code>}</code>	
<code>}</code>	

Como se muestra en la Tabla 4, el codificador de vídeo 20 está configurado para señalar una tabla de tamaños de DPB que incluye los elementos sintácticos cuyos valores especifican un tamaño de sub-DPB (`max_vps_dec_pic_buffering_minus1[i][k][j]`), el número máximo de imágenes reordenadas (`max_vps_num_reorder_pics[i][j]`), y la latencia máxima de sub-DPB (`max_vps_latency_increase_plus1[i][j]`). El bucle for "for(i = 1; i < NumOutputLayerSets; i++)" especifica que el tamaño de sub-DPB, el número máximo de imágenes reordenadas y la latencia máxima de sub-DPB se señalizan para cada conjunto de capas de salida i (es decir, para cada punto de operación) hasta el número máximo de conjuntos de capas de salida (NumOutputLayerSets). Por consiguiente, el descodificador de vídeo 30 está configurado para aplicar el tamaño máximo de sub-DPB recibido, el número máximo de imágenes reordenadas y la latencia máxima de sub-DPB para cada capa en un conjunto de capas de salida de un punto de operación.

[0112] Además, como se muestra en la Tabla 4, el codificador de vídeo 20 también puede configurarse para generar y señalar un indicador (`sub_layer_flag_info_present_flag[i]`) que indica si los valores señalizados indicativos del tamaño máximo de sub-DPB, el número máximo de imágenes reordenadas y la latencia máximo de sub-DPB se señalizan solamente para una subcapa de cada capa, o para más de una subcapa de una capa. Por ejemplo, si el valor de `sub_layer_flag_info_present_flag[i]` es 1, los valores de los elementos sintácticos que indican el tamaño máximo de sub-DPB, el número máximo de imágenes reordenadas y la latencia máxima de sub-DPB se utilizarán para cada subcapa. Sin embargo, si el valor de `sub_layer_flag_info_present_flag[i]` es 0, los elementos sintácticos separados cuyos valores indican el tamaño máximo de sub-DPB, el número máximo de imágenes reordenadas y la latencia máxima de sub-DPB pueden señalizarse para más de una subcapa. En este ejemplo, el descodificador de vídeo 30, basado en el valor de los elementos sintácticos respectivos, puede configurarse para usar diferentes valores para el tamaño máximo de sub-DPB, el número máximo de imágenes reordenadas y la latencia máxima de sub-DPB para dos o más subcapas diferentes.

[0113] En el caso de que el valor de `sub_layer_flag_info_present_flag[i]` sea 1 (lo que indica que los parámetros de sub-DPB se señalarán para más de una subcapa), el codificador de vídeo 20 puede configurarse adicionalmente para señalar un elemento sintáctico adicional (`sub_layer_dpb_info_present_flag[i][j]`) para cada subcapa en cada sub-DPB que indica si o no los parámetros de sub-DPB serán señalizados para una subcapa particular.

[0114] En el ejemplo de la Tabla 4, la instrucción if ("`if(j > 0 && sub_layer_flag_info_present_flag[i] && sub_layer_dpb_info_present_flag[i][j-1])`") requiere un codificador de vídeo 20 para señalar el `sub_layer_dpb_info_present_flag[i][j]` para una subcapa particular solamente si el valor de `sub_layer_flag_info_present_flag[i]` es 1 y el valor `sub_layer_dpb_info_present_flag` de la subcapa anterior (`sub_layer_dpb_info_present_flag[i][j-1]`) tiene un valor de 1. Es decir, el codificador de vídeo 20 no señala el `sub_layer_dpb_info_present_flag`, ni ningún parámetro de sub-DPB, si los parámetros de sub-DPB no se han señalado para una subcapa anterior.

[0115] En un ejemplo de la divulgación, se elimina el requisito de verificar el valor del `sub_layer_dpb_info_present_flag` para una subcapa anterior. En su lugar, el `sub_layer_dpb_info_present_flag` puede señalizarse para cada subcapa, independientemente de los valores anteriores del `sub_layer_dpb_info_present_flag`.

[0116] La semántica asociada con los elementos sintácticos de la Tabla 4 anterior se describe a continuación. Como se usa en el presente documento, `NumSubDpbs[i]` puede denotar un número de sub-DPB requeridos para un i-ésimo conjunto de capas de salida.

[0117] El valor del elemento sintáctico `sub_layer_flag_info_present_flag[i]` igual a 1 indica que `sub_layer_dpb_info_present_flag[i][j]` está presente para todas las subcapas en el i-ésimo conjunto de capas de salida. El número máximo de subcapas se indica mediante el valor del elemento sintáctico `vps_max_sub_layers_minus1`. En otras palabras, cuando el codificador de vídeo 20 genera un `sub_layer_flag_info_present_flag[i]` con un valor de 1, el descodificador de vídeo 30 determina que se señalará un indicador adicional (es decir, `sub_layer_dpb_info_present_flag[i][j]`) para todas las subcapas del i-ésimo conjunto de capas de salida.

[0118] Por otro lado, si el codificador de vídeo 20 genera un elemento sintáctico `sub_layer_flag_info_present_flag[i]` con un valor igual a 0, el descodificador de vídeo 30 puede determinar que el

elemento sintáctico `sub_layer_flag_info_present_flag[i]` especifica que, para cada valor de `j` mayor que 0, `sub_layer_dpb_info_present_flag[i][j]` no está presente, y el descodificador de vídeo 30 puede inferir que el valor sea igual a 0. Como tal, el codificador de vídeo 20 usará los valores inicialmente señalizados de los parámetros de sub-DPB (es decir, `max_vps_dec_pic_buffering_minus1[i][k][j]`, `max_vps_num_reorder_pics[i][j]` y `max_vps_latency_increase_plus1[i][j]`) para cada subcapa.

[0119] Como se ha descrito anteriormente, si el codificador de vídeo 20 genera el `sub_layer_flag_info_present_flag[i]` con un valor de 1, el codificador de vídeo 20 también genera un `sub_layer_dpb_info_present_flag[i][j]` para cada subcapa de un conjunto de capas de salida. Si el valor del elemento sintáctico `sub_layer_dpb_info_present_flag[i][j]` es igual a 1, entonces el descodificador de vídeo 30 puede recibir el tamaño de sub-DPB (`max_vps_dec_pic_buffering_minus1[i][k][j]`), para la subcapa `j` para cada sub-DPB `k` en el intervalo de 0 al número máximo de sub-DPB (`NumSubDpbs[i]-1`), inclusive. El descodificador de vídeo 30 está configurado para recibir también elementos sintácticos que indican la latencia máxima (`max_vps_latency_increase_plus1[i][j]`) y el número máximo de imágenes reordenadas (`max_vps_num_reorder_pics[i][j]`) para la subcapa `j`.

[0120] Por otro lado, si el descodificador de vídeo 30 determina que el valor del elemento sintáctico `sub_layer_dpb_info_present_flag[i][j]` es igual a 0, entonces el descodificador de vídeo 30 puede determinar que el elemento sintáctico `sub_layer_dpb_info_present_flag[i][j]` especifica que los valores de `max_vps_dec_pic_buffering_minus1[i][k][j]` son iguales a `max_vps_dec_pic_buffering_minus1[i][k][j-1]` (es decir, igual al valor del elemento sintáctico de tamaño de DPB para la subcapa anterior en el conjunto de capas de salida) para `k` en el intervalo de 0 a (`NumSubDpbs[i]-1`), inclusive. Además, el descodificador de vídeo 30 puede determinar además que los valores del número máximo del elemento sintáctico de imágenes reordenadas (`max_vps_num_reorder_pics[i][j]`) y el elemento sintáctico de latencia máxima (`max_vps_latency_increase_plus1[i][j]`) para la subcapa `j` se establece para que sean iguales al valor de esos elementos sintácticos para la subcapa anterior en el conjunto de capas de salida (es decir, `max_vps_num_reorder_pics[i][j-1]` y `max_vps_latency_increase_plus1[i][j-1]`, respectivamente). Además, el descodificador de vídeo 30 puede inferir el valor del elemento sintáctico `sub_layer_dpb_info_present_flag[i]` para que cualquier valor posible de `i` sea igual a 1. Es decir, el descodificador de vídeo 30 puede estar configurado para inferir un valor de 1 del `sub_layer_dpb_info_present_flag` en la 0^{ésima} (es decir, la primera) subcapa de un conjunto de capas de salida. Además, el descodificador de vídeo 30 puede inferir que el valor del elemento sintáctico `sub_layer_dpb_info_present_flag[i][k]` es igual a 0 para `k` en el intervalo de `j + 1` hasta `vps_max_sub_layers_minus1`, inclusive.

[0121] El elemento sintáctico `max_vps_dec_pic_buffering_minus1[i][k][j]` más 1 especifica el tamaño máximo requerido del `k`-ésimo sub-DPB para el CVS en el `i`-ésimo conjunto de capas de salida en unidades de búferes de almacenamiento de imágenes cuando `HighestTid` es igual a `j`. Cuando `j` es mayor que 0, el descodificador de vídeo 30 puede determinar que `max_vps_dec_pic_buffering_minus1[i][k][j]` es mayor o igual que el valor de `max_vps_dec_pic_buffering_minus1[i][k][j-1]`. Cuando `max_vps_dec_pic_buffering_minus1[i][k][j]` no está presente para `j` en el intervalo de 1 a (`vps_max_sub_layers_minus1-1`), inclusive, el descodificador de vídeo 30 puede inferir el valor de `max_vps_dec_pic_buffering_minus1[i][k][j]` para que sea igual al valor de `max_vps_dec_pic_buffering_minus1[i][k][j-1]`.

[0122] El elemento sintáctico `max_vps_num_reorder_pics[i][j]` indica el número máximo de imágenes reordenadas. Es decir, el número máximo permitido de **unidades de acceso que contienen una imagen con PicOutputFlag igual a 1** que pueden preceder a cualquier **unidad de acceso auA que contenga una imagen con PicOutputFlag igual a 1** en el `i`-ésimo conjunto de capas de salida en el CVS en el orden de decodificación e ir detrás de la **unidad de acceso auA que contiene una imagen con PicOutputFlag igual a 1** en el orden de salida, cuando `HighestTid` es igual a `j`. Cuando `max_vps_num_reorder_pics[i][j]` no está presente para `j` en el intervalo de 1 a (`vps_max_sub_layers_minus1-1`), inclusive, debido a que `sub_layer_dpb_info_present_flag[i][j]` es igual a 0, el descodificador de vídeo 30 puede inferir el valor del elemento sintáctico `max_vps_num_reorder_pics[i][j]` para que sea igual al valor de `max_vps_num_reorder_pics[i][j-1]`.

[0123] El valor del elemento sintáctico `max_vps_latency_increase_plus1[i][j]` indica la latencia máxima. El elemento sintáctico `max_vps_latency_increase_plus1[i][j]` no igual a 0 se usa para calcular el valor del elemento sintáctico `VpsMaxLatencyPictures[i][k][j]`, cuyo valor especifica el número máximo de **unidades de acceso que contienen una imagen con PicOutputFlag igual a 1** en el `i`-ésimo conjunto de capas de salida que puede preceder a cualquier **unidad de acceso auA que contiene una imagen con PicOutputFlag igual a 1** en el CVS en el orden de salida e ir detrás de **la unidad de acceso auA que contiene una imagen con PicOutputFlag igual a 1** en el orden de decodificación cuando `HighestTid` es igual a `j`. En los casos en que `max_vps_latency_increase_plus1[i][j]` no está presente para `j` en el intervalo de 1 a (`vps_max_sub_layers_minus1-1`), inclusive, debido a que `sub_layer_dpb_info_present_flag[i][j]` es igual a 0, el descodificador de vídeo 30 puede inferir el valor de `max_vps_latency_increase_plus1[i][j]` para que sea igual a `max_vps_latency_increase_plus1[i][j-1]`.

[0124] Cuando `max_vps_latency_increase_plus1[i][j]` no es igual a 0, el codificador de vídeo 20 puede especificar el valor de `VpsMaxLatencyPictures[i][k][j]` de la siguiente manera:

$$\text{VpsMaxLatencyPictures}[i][j] = \text{max_vps_num_reorder_pics}[i][j] + \text{max_vps_latency_increase_plus1}[i][j] - 1$$

Es decir, el número máximo de imágenes de latencia es igual al número máximo de imágenes reordenadas más el valor del elemento sintáctico de latencia máxima.

[0125] Cuando `max_vps_latency_increase_plus1[i][j]` es igual a 0, no se expresa ningún límite correspondiente. El valor de `max_vps_latency_increase_plus1[i][j]` estará en el intervalo de 0 a $[(2^{32})-2]$, inclusive.

[0126] A continuación se muestran los cambios en el proceso de salida y eliminación de imágenes en relación con la subcláusula F.13.5.2.2 de HEVC con el fin de implementar las técnicas de ejemplo de esta divulgación. Nuevamente, las adiciones se muestran en **negrita y subrayado**, y las supresiones se muestran con a.

[0127] La salida y eliminación de imágenes del DPB antes de la descodificación de la imagen actual por el descodificador de vídeo 30 (pero después de analizar sintácticamente la cabecera de fragmento del primer fragmento de la imagen actual) se produce instantáneamente cuando la primera unidad de descodificación de la imagen actual se elimina del CPB y procede de la siguiente manera. Se invoca el proceso de descodificación para RPS como se especifica en la subcláusula F.8.3.2.

- Si la imagen actual es una imagen IRAP con `NoRaslOutputFlag` igual a 1 y con `nuh_layer_id` igual a 0 que no es la imagen 0, el descodificador de vídeo 30 puede aplicar las siguientes etapas ordenadas:

1. El descodificador de vídeo 30 puede derivar la variable `NoOutputOfPriorPicsFlag` para el descodificador sometido a prueba de la siguiente manera:

- Si la imagen actual es una imagen CRA, el descodificador de vídeo 30 puede establecer el `NoOutputOfPriorPicsFlag` igual a 1 (independientemente del valor de `no_output_of_prior_pics_flag`).

- En caso contrario, si el valor de `pic_width_in_luma_samples`, `pic_height_in_luma_samples`, o `sps_max_dec_pic_buffering_minus1[HighestTid]` derivado del SPS activo es diferente del valor de `pic_width_in_luma_samples`, `pic_height_in_luma_samples`, o `sps_max_dec_pic_buffering_minus1[HighestTid]`, respectivamente, derivado del SPS activo para la imagen precedente en el orden de descodificación con el mismo valor de `nuh_layer_id` que la imagen actual, el descodificador de vídeo 30 puede establecer `NoOutputOfPriorPicsFlag` en 1 sometido a prueba, independientemente del valor de `no_output_of_prior_pics_flag`. Sin embargo, en los ejemplos preferentes, el descodificador de vídeo 30 no puede establecer `NoOutputOfPriorPicsFlag` en 1 sometido a prueba en las condiciones especificadas en este punto. NOTA - Aunque puede ser preferente establecer `NoOutputOfPriorPicsFlag` igual a `no_output_of_prior_pics_flag` en estas condiciones, se permite que el descodificador de vídeo 30, sometido a prueba, establezca `NoOutputOfPriorPicsFlag` en 1 en estas condiciones.

- En caso contrario, el descodificador de vídeo 30 puede establecer `NoOutputOfPriorPicsFlag` igual a `no_output_of_prior_pics_flag`.

2. El descodificador de vídeo 30 puede aplicar el valor de `NoOutputOfPriorPicsFlag` derivado sometido a prueba para el HRD de la siguiente manera:

- Si `NoOutputOfPriorPicsFlag` es igual a 1, el descodificador de vídeo 30 puede vaciar todos los búferes de almacenamiento de imágenes en todos los sub-DPB en el DPB sin salida de las imágenes contenidas en los búferes de almacenamiento de imágenes vaciados, y el descodificador de vídeo 30 puede establecer la plenitud del sub-DPB igual a 0 para todos los sub-DPB.

- En caso contrario (es decir, `NoOutputOfPriorPicsFlag` es igual a 0), el descodificador de vídeo 30 puede vaciar todos los búferes de almacenamiento de imágenes que contienen una imagen que está marcada como "no necesaria para la salida" y "no utilizada de referencia" sin salida, y el descodificador de vídeo 30 puede vaciar todos los búferes de almacenamiento de imágenes que no están vacíos en el DPB invocando repetidamente el proceso de "expulsión" especificado en la subcláusula F.13.5.2.4, y el descodificador de vídeo 30 puede establecer la plenitud del DPB igual a 0.

- En caso contrario (por ejemplo, si la imagen actual no es una imagen IRAP con `NoRaslOutputFlag` igual a 1 o con un id de capa `nuh` no igual a 0), el descodificador de vídeo 30 puede vaciar todos los búferes de almacenamiento de imágenes que contienen una imagen que está marcada como "no necesaria para la salida" y

"no utilizada de referencia" sin salida. Para cada búfer de almacenamiento de imágenes que se vacía, el descodificador de vídeo 30 puede disminuir la plenitud de sub-DPB del sub-DPB asociado con ese búfer de almacenamiento de imágenes en uno. La variable currLayerId se establece igual a nuh_layer_id de la imagen descodificada actual, la variable currSubDpbId se establece igual al índice del sub-DPB asociado con la imagen descodificada actual, y cuando una o más de las siguientes condiciones son verdaderas, el proceso de "expulsión" especificado en la subcláusula F.13.5.2.4 se invoca repetidamente mientras se disminuye aún más la plenitud de DPB en uno por cada búfer de almacenamiento de imágenes adicional que se vacía, hasta que ninguna de las siguientes condiciones sea verdadera:

10 - El número de **unidades de acceso que contienen al menos una imagen** que **está** marcada como "necesaria para la salida" es mayor que sps_max_num_reorder_pics[HighestTid] del SPS activo (**cuando una secuencia de vídeo codificada que se ajusta a uno o más de los perfiles especificados en el Anexo A se decodifica aplicando el proceso de descodificación especificado en las cláusulas 2-10)** o max_vps_num_reorder_pics[TargetOutputLayerId] [HighestTid] desde el VPS de la capa activa para el valor de currLayerId (**cuando una secuencia de vídeo codificada que se ajusta a uno o más de los perfiles especificados en el Anexo G o H se decodifica aplicando el proceso de descodificación especificado en las cláusulas 2-10, Anexo F y Anexo G o H).**

20 - sps_max_latency_increase_plus1[HighestTid] del SPS activo (**cuando una secuencia de vídeo codificada que se ajusta a uno o más de los perfiles especificados en el Anexo A se decodifica aplicando el proceso de descodificación especificado en las cláusulas 2-10)** o VpsMaxLatencyPictures[TargetOutputLayerId] [HighestTid] del VPS activo (**cuando una secuencia de vídeo codificada que se ajusta a uno o más de los perfiles especificados en el Anexo G o H se decodifica aplicando el proceso de descodificación especificado en las cláusulas 2-10, Anexo F y Anexo G o H)**) no es igual a 0 y hay al menos una **unidad de acceso que contiene una imagen que está marcada como "necesaria para la salida"** en el DPB para la cual la variable asociada PicLatencyCount[currLayerId] es mayor o igual que SpsMaxLatencyPictures[HighestTid] derivada del SPS activo (**cuando una secuencia de vídeo codificada conforme a uno o más de los perfiles especificados en el Anexo A se decodifica aplicando el proceso de descodificación especificado en las cláusulas 2-10)** o VpsMaxLatencyPictures[TargetOutputLayerId] [HighestTid] del VPS activo para el valor de currLayerId (**cuando una secuencia de vídeo codificada que se ajusta a uno o más de los perfiles especificados en el Anexo G o H se decodifica aplicando el proceso de descodificación especificado en las cláusulas 2-10, Anexo F y Anexo G o H).**

35 - El número de imágenes con id de capa nuh igual a currLayerId en el sub-DPB asociado es mayor o igual que sps_max_dec_pic_buffering_minus1[HighestTid] + 1 del SPS activo (**cuando una secuencia de vídeo codificada se ajusta a uno o más de los perfiles especificados en el Anexo A se decodifica aplicando el proceso de descodificación especificado en las cláusulas 2-10)** o max_vps_dec_pic_buffering_minus1[TargetOutputLayerId][currSubDpbId][HighestTid] + 1 (**cuando se decodifica una secuencia de vídeo codificada que se ajusta a uno o más de los perfiles especificados en el Anexo G o H aplicando el proceso de descodificación especificado en las cláusulas 2-10, Anexo F y Anexo G o H).**

45 **[0128]** Como se muestra anteriormente en las adiciones y supresiones marcadas, el proceso de salida y eliminación de imágenes (también llamado "expulsión") se actualiza para reflejar las técnicas de señalización de parámetros de sub-DPB de esta divulgación. Un cambio es que las condiciones para continuar el proceso de expulsión se basan en unidades de acceso en lugar de imágenes. Esto se debe a que, de acuerdo con las técnicas de esta divulgación, los parámetros de sub-DPB se aplican a una capa de salida. También en este sentido, se elimina la necesidad de verificar los valores de los parámetros de sub-DPB para los ID de capa particulares (por ejemplo, LayerIdxInVps y currLayerId), ya que cada capa de un conjunto de capas de salida particular comparte esos mismos parámetros de sub-DPB.

55 **[0129]** A continuación se muestran los cambios en los procesos de descodificación, marcado, expulsión adicional y almacenamiento en relación con la subcláusula HEVC C.1.1.1,1 de HEVC a fin de implementar las técnicas de ejemplo de esta divulgación. Nuevamente, las adiciones se muestran en **negrita y subrayado**, y las supresiones se muestran con a.

60 **[0130]** Los procesos especificados en esta subcláusula se producen instantáneamente cuando el codificador de vídeo 20 y/o el descodificador de vídeo 30 eliminan la última unidad de descodificación de la unidad de acceso n que contiene la imagen actual del CPB. El codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden establecer la variable currLayerId igual a nuh_layer_id de la imagen descodificada actual, y pueden establecer currSubDpbId igual al índice del sub-DPB asociado con la capa a la que está asociada la imagen actual. Los cambios introducidos por las técnicas de esta divulgación en los procesos especificados de descodificación de imágenes, marcado, expulsión adicional y almacenamiento se denotan a continuación utilizando texto subrayado y tachado.

65

[0131] Para la unidad de acceso que contiene una imagen en el sub-DPB asociado que está marcada como "necesaria para la salida", el codificador de vídeo 20 y/o el descodificador de vídeo 30 pueden establecer la variable asociada PicLatencyCount igual a PicLatencyCount + 1.

5 **[0132] El descodificador de vídeo 30 puede considerar la imagen actual como descodificada después de que se descodifique la última unidad de descodificación de la imagen. La imagen descodificada actual puede almacenarse en un búfer de almacenamiento de imágenes vacío en el DPB, y puede aplicarse lo siguiente:**

10 - **Si la imagen descodificada actual tiene PicOutputFlag igual a 1, el descodificador de vídeo 30 puede marcar la imagen descodificada actual como "necesaria para la salida" y la variable PicLatencyCount asociada con la unidad de acceso actual se establece igual a 0.**

15 - **En caso contrario (por ejemplo, la imagen descodificada actual tiene PicOutputFlag igual a 0), el descodificador de vídeo 30 puede marcar la imagen descodificada actual como "no necesaria para la salida".**

[0133] Además, el descodificador de vídeo 30 puede marcar la imagen descodificada actual como "utilizada de referencia a corto plazo".

20 **[0134]** Cuando una o más de las siguientes condiciones son verdaderas, el descodificador de vídeo 30 puede invocar repetidamente/iterativamente el proceso de "expulsión" especificado en la subcláusula F.13.5.2.4, hasta que ninguna de las siguientes condiciones sea verdadera:

25 - El **número de unidades de acceso que contienen al menos una imagen** que está marcada como "necesaria para la salida" **es** mayor que `sps_max_num_reorder_pics[HighestTid]` del SPS activo **cuando una secuencia de vídeo codificada que se ajusta a uno o más de los perfiles especificados en el Anexo A se decodifica aplicando el proceso de descodificación especificado en las cláusulas 2-10)** o `max_vps_num_reorder_pics[TargetOutputLayerId][HighestTid]` desde el VPS activo para el valor de `currLayerId` **(cuando una secuencia de vídeo codificada que se ajusta a uno o más de los perfiles especificados en el Anexo G o H se decodifica aplicando el proceso de descodificación especificado en las cláusulas 2-10, Anexo F y Anexo G o H).**

30 - `sps_max_latency_increase_plus1[HighestTid]` no es igual a 0 **cuando una secuencia de vídeo codificada que se ajusta a uno o más de los perfiles especificados en el Anexo A se decodifica aplicando el proceso de descodificación especificado en las cláusulas 2-10)** o `VpsMaxLatencyPictures[TargetOutputLayerId][HighestTid]` del VPS activo **(cuando una secuencia de vídeo codificada que se ajusta a uno o más de los perfiles especificados en el Anexo G o H se decodifica aplicando el proceso de descodificación especificado en las cláusulas 2-10, Anexo F y Anexo G o H)** no es igual a 0 y hay al menos **una unidad de acceso que contiene una imagen marcada como "necesaria para la salida"** para la cual la variable asociada `PicLatencyCount` es mayor o igual que `SpsMaxLatencyPictures[HighestTid]` derivada del SPS activo **(cuando una secuencia de vídeo codificada conforme a uno o más de los perfiles especificados en el Anexo A se decodifica aplicando el proceso de descodificación especificado en las cláusulas 2-10)** o `VpsMaxLatencyPictures[TargetOutputLayerId][HighestTid]` del VPS activo para el valor de `currLayerId` **(cuando una secuencia de vídeo codificada que se ajusta a uno o más de los perfiles especificados en el Anexo G o H se decodifica aplicando el proceso de descodificación especificado en las cláusulas 2-10, Anexo F y Anexo G o H).**

50 **[0135]** Como se muestra anteriormente en las adiciones y supresiones marcadas, el proceso de salida y eliminación de imágenes (también llamado "expulsión") se actualiza para reflejar las técnicas de señalización de parámetros de sub-DPB de esta divulgación. Un cambio es que las condiciones para continuar el proceso de expulsión se basan en unidades de acceso en lugar de imágenes. Esto se debe a que, de acuerdo con las técnicas de esta divulgación, los parámetros de sub-DPB se aplican a un conjunto completo de capas de salida y no solo a un subconjunto de las capas.

55 **[0136]** Los párrafos siguientes son una descripción de una segunda implementación de ejemplo en relación con la primera implementación de ejemplo descrita anteriormente. A diferencia de la implementación del primer ejemplo, de acuerdo con la implementación del segundo ejemplo, el codificador de vídeo 20 puede indicar un ID de capa que especifica la capa con la que están asociados los valores de `max_vps_num_reorder_pics[i][j]` y `max_vps_latency_increase_plus1[i][j]`. Los cambios en la sintaxis y la semántica introducidos por la implementación del segundo ejemplo, se distinguen a continuación con **negrita y subrayado** y formateando, como se ha mostrado anteriormente. La sintaxis y la semántica de los otros elementos sintácticos (es decir, indicados por texto que no está subrayado ni tachado) siguen siendo los mismos, de acuerdo con la implementación del segundo ejemplo

65

Tabla 5 - Sintaxis de la tabla de tamaños de DPB

dpb_size_table() {	
for(i = 1; i < NumOutputLayerSets; i++) {	
sub_layer_flag_info_present_flag[i]	u(1)
for(j = 0; j <= vps_max_sub_layers_minus1; j++) {	
if(j > 0 && sub_layer_flag_info_present_flag[i] &&	
sub_layer_dpb_info_present_flag[i][j] - 1)	
sub_layer_dpb_info_present_flag [i][j]	u(1)
if(sub_layer_dpb_info_present_flag[i][j]) {	
for(k = 0; k < NumSubDpbs[i]; k++)	
max_vps_dec_pic_buffering_minus1[i][k][j]	ue(v)
reorder_info_layer_id[i][j]	
max_vps_num_reorder_pics[i][j]	ue(v)
max_vps_latency_increase_plus1[i][j]	ue(v)
}	
}	
}	
}	

[0137] La Tabla 5 anterior describe los cambios de sintaxis introducidos por la implementación del segundo ejemplo en relación con la implementación del primer ejemplo. Los cambios en la semántica, tal como se presentan en la implementación del segundo ejemplo, se describen a continuación.

[0138] El descodificador de vídeo 30 puede determinar que el elemento sintáctico **reorder_info_layer_id[i][j]** especifica el valor nuh_layer_id de la capa que pertenece al i-ésimo conjunto de capas de salida al cual se aplican los valores de max_vps_num_reorder_pics[i][j] y max_vps_latency_increase_plus1[i][j]. En los casos en que el elemento sintáctico reorder_info_layer_id[i][j] no está presente, el descodificador de vídeo 30 puede inferir que el valor de reorder_info_layer_id[i][j] es igual a 0. De forma alternativa, en algunos ejemplos, cuando reorder_info_layer_id[i][j] no está presente, el descodificador de vídeo 30 puede inferir que el valor de reorder_info_layer_id[i][j] es igual al nuh_layer_id de la capa más alta en el i-ésimo conjunto de capas de salida. De forma alternativa, en algunos ejemplos, el codificador de vídeo 20 puede señalar un indicador que especifica al descodificador de vídeo 30 si se señala el elemento sintáctico reorder_info_layer_id[i][j].

[0139] El descodificador de vídeo 30 puede determinar que el elemento sintáctico **max_vps_num_reorder_pics[i][j]** indica el número de imágenes máximo permitido con nuh_layer_id igual a **reorder_info_layer_id[i][j]** que puede preceder a cualquier imagen con id de capa nuh igual a **reorder_info_layer_id[i][j]** en el i-ésimo conjunto de capas de salida en el CVS en el orden de descodificación, e ir detrás de esa imagen con nuh_layer_id igual a **reorder_info_layer_id[i][j]** en el orden de salida, y cuando HighestTid es igual a j. Cuando max_vps_num_reorder_pics[i][j] no está presente para j en el intervalo de 1 a (vps_max_sub_layers_minus1-1), inclusive, debido a que sub_layer_dpb_info_present_flag[i][j] es igual a 0, el descodificador de vídeo 30 puede inferir el valor del elemento sintáctico max_vps_num_reorder_pics[i][j] para que sea igual a max_vps_num_reorder_pics[i][j-1].

[0140] En los casos en que el valor de **max_vps_latency_increase_plus1[i][j]** no es igual a 0, el descodificador de vídeo 30 puede usar el valor del elemento sintáctico max_vps_latency_increase_plus1[i][j] para calcular el valor de VpsMaxLatencyPictures[i][k][j]. El valor del elemento sintáctico VpsMaxLatencyPictures[i][k][j], a su vez, puede especificar el número máximo de imágenes con nuh_layer_id igual a **reorder_info_layer_id[i][j]** en el i-ésimo conjunto de capas de salida que puede preceder a cualquier imagen con id de capa nuh igual a **reorder_info_layer_id[i][j]** en el CVS en el orden de salida e ir detrás de esa imagen con id de capa nuh igual a **reorder_info_layer_id[i][j]** en el orden de descodificación cuando HighestTid es igual a j. En los ejemplos que max_vps_latency_increase_plus1[i][j] no está presente para j en el intervalo de 1 a (vps_max_sub_layers_minus1-1), inclusive, debido a que el valor de sub_layer_dpb_info_present_flag[i][j] es igual a 0, el descodificador de vídeo 30 puede inferir el valor de max_vps_latency_increase_plus1[i][j] para que sea igual a max_vps_latency_increase_plus1[i][j-1].

[0141] En los casos en que max_vps_latency_increase_plus1[i][j] no es igual a 0, el valor de VpsMaxLatencyPictures[i][k][j] se especifica mediante una fórmula de la siguiente manera:

$$\text{VpsMaxLatencyPictures}[i][j] = \text{max_vps_num_reorder_pics}[i][j] + \text{max_vps_latency_increase_plus1}[i][j] - 1$$

5 **[0142]** Por el contrario, en los ejemplos en los que `max_vps_latency_increase_plus1[i][j]` es igual a 0, no se expresa ningún límite correspondiente. En estos ejemplos, el codificador de vídeo 20 y/o el decodificador de vídeo 30 pueden determinar que el valor de `max_vps_latency_increase_plus1[i][j]` está en el intervalo de 0 a $(2^{32}-2)$, inclusive.

10 **[0143]** A continuación se describe un tercer ejemplo de implementación relativo al primer ejemplo de implementación. Sin embargo, a diferencia de la implementación del primer ejemplo, de acuerdo con la implementación del tercer ejemplo, el codificador de vídeo 20 puede señalar un indicador para especificar si el número máximo de imágenes reordenadas y la latencia máxima se señalizan para cada capa en el conjunto de capas de salida. Por ejemplo, de acuerdo con la implementación del tercer ejemplo, el codificador de vídeo 20 puede establecer el indicador en un primer valor para indicar que el número máximo de imágenes reordenadas y la latencia máxima se señalizan para cada capa en el conjunto de capas de salida, y en un segundo valor para indicar que el número máximo de imágenes reordenadas y la latencia máxima no se señalizan para cada capa en el conjunto de capas de salida. Los cambios introducidos por la implementación del tercer ejemplo, en comparación con la sintaxis proporcionada por la implementación del primer ejemplo, se distinguen **subrayando y en negrita** en la Tabla 6 a continuación.

Tabla 6 - Sintaxis de la tabla de tamaños de DPB

<code>dpb_size_table() {</code>	
<code>for(i = 1; i < NumOutputLayerSets; i++) {</code>	
<u>sub_layer_flag_info_present_flag[i]</u>	u(1)
<code>for(j = 0; j <= vps_max_sub_layers_minus1; j++) {</code>	
<code>if(j > 0 && sub_layer_flag_info_present_flag[i] &&</code>	
<code>sub_layer_dpb_info_present_flag[i][j - 1])</code>	
<u>sub_layer_dpb_info_present_flag[i][j]</u>	u(1)
<code>if(sub_layer_dpb_info_present_flag[i][j]) {</code>	
<code>for(k = 0; k < NumSubDpbs[i]; k++)</code>	
<u>max_vps_dec_pic_buffering_minus1[i][k][i]</u>	ue(v)
<u>reorder_info_only_highest_layer_flag[i][j]</u>	u(1)
<code>for(k = reorder_info_only_highest_layer_flag[i][j] ?</code>	
<code>NumLayersInIdList[output_layer_set_idx_minus1[i]+1]-1;0;</code>	
<code>k <= NumLayersInIdList[output_layer_set_idx_minus1[i]+1]-1;</code>	
<code>k++) {</code>	
<u>max_vps_num_reorder_pics[i][k][i]</u>	ue(v)
<u>max_vps_latency_increase_plus1[i][k][i]</u>	ue(v)
<code>}</code>	
<code>}</code>	
<code>}</code>	
<code>}</code>	

25 **[0144]** A continuación se describe una nueva semántica introducida por la implementación del tercer ejemplo con respecto a la tabla de tamaños DPB de la Tabla 6, en comparación con la implementación del primer ejemplo. Cualquier semántica que no se describa a continuación con respecto a la implementación del tercer ejemplo sigue siendo la misma que en la implementación del primer ejemplo.

30 **[0145]** En los ejemplos en los que el codificador de vídeo 20 establece el `sub_layer_flag_info_present_flag[i]` igual a 1, el `sub_layer_flag_info_present_flag[i]` especifica que `sub_layer_dpb_info_present_flag[i][j]` está presente para las subcapas `vps_max_sub_layers_minus1`. Por el contrario, en los ejemplos en los que el codificador de vídeo 20 establece `sub_layer_flag_info_present_flag[i]` igual a 0, el `sub_layer_flag_info_present_flag[i]` especifica que, para cada valor de `j` mayor que 0, `sub_layer_dpb_info_present_flag[i][j]` no está presente. En algunos ejemplos

donde el codificador de vídeo 20 establece el `sub_layer_flag_info_present_flag[i]` en 0, el descodificador de vídeo 30 puede inferir que el valor de `sub_layer_dpb_info_present_flag[i][j]` es igual a 0.

5 **[0146]** En los ejemplos en los que el codificador de vídeo establece el `sub_layer_dpb_info_present_flag[i][j]` igual a 1, el `sub_layer_dpb_info_present_flag[i][j]` especifica el descodificador de vídeo 30 que `max_vps_dec_pic_buffering_minus1[i][k][j]`, para k en el intervalo de 0 a $(\text{NumSubDpbs}[i]-1)$, inclusive, y `reorder_info_only_highest_layer_flag[i][j]` están presentes para la subcapa j -ésima. Por otro lado, en los ejemplos en los que el codificador de vídeo 20 establece el `sub_layer_dpb_info_present_flag[i][j]` igual a 0, el `sub_layer_dpb_info_present_flag[i][j]` especifica al descodificador de vídeo 30 que los `max_vps_dec_pic_buffering_minus1[i][k][j]` son iguales a `max_vps_dec_pic_buffering_minus1[i][k][j-1]` para k en el intervalo de 0 a $(\text{NumSubDpbs}[i]-1)$, inclusive, y que los valores `max_vps_num_reorder_pics[i][k][j]` y `max_vps_latency_increase_plus1[i][k][j]` se establecen iguales a `max_vps_num_reorder_pics[i][k][j-1]` y `max_vps_latency_increase_plus1[i][k][j-1]`, respectivamente, para k en el intervalo de 0 a $(\text{NumLayersInIdList}[\text{output_layer_set_idx_minus1}[i+1]-1])$, inclusive. El descodificador de vídeo 30 puede inferir el valor de `sub_layer_dpb_info_present_flag[i]` para que cualquier valor posible de i sea igual a 1.

20 **[0147]** El valor de `max_vps_dec_pic_buffering_minus1[i][k][j]` más 1 especifica el tamaño máximo requerido del k -ésimo sub-DPB para el CVS en el i -ésimo conjunto de capas de salida en unidades de búferes de almacenamiento de imágenes cuando `HighestTid` es igual a j . Para cualquier k mayor que 0, `max_vps_dec_pic_buffering_minus1[i][k][j]` será menor o igual que `max_vps_dec_pic_buffering_minus1[i][k][j-1]`. Cuando `max_vps_dec_pic_buffering_minus1[i][k][j]` no está presente para j en el intervalo de 1 a $(\text{vps_max_sub_layers_minus1} - 1)$, inclusive, el descodificador de vídeo 30 puede inferir `max_vps_dec_pic_buffering_minus1[i][k][j]` para que sea igual a `max_vps_dec_pic_buffering_minus1[i][k][j-1]`.

25 **[0148]** En los ejemplos en los que `reorder_info_only_highest_layer_flag[i][j]` es igual a 1, `reorder_info_only_highest_layer_flag[i][j]` indica al descodificador de vídeo 30 que los elementos sintácticos `max_vps_num_reorder_pics[i][NumLayersInIdList[output_layer_set_idx_minus1[i] + 1]-1][j]` y `max_vps_latency_increase_plus1[i][NumLayersInIdList[output_layer_set_idx_minus1[i+1]-1][j]` están presentes y `max_vps_num_reorder_pics[i][k][j]` y `max_vps_latency_increase_plus1[i][k][j]` para k en el intervalo de 0 a $(\text{NumLayersInIdList}[\text{output_layer_set_idx_minus1}[i+1]-2])$, inclusive, se infiere (por ejemplo, mediante el descodificador de vídeo 30) para que sea igual a `max_vps_num_reorder_pics[i][NumLayersInIdList[output_layer_set_idx_minus1[i] + 1]-1][j]` y `max_vps_latency_increase_plus1[i][NumLayersInIdList[output_layer_set_idx_minus1[i+1]-1][j]`, respectivamente. Por otro lado, en los ejemplos en los que `reorder_info_only_highest_layer_flag[i][j]` es igual a 0, el `reorder_info_only_highest_layer_flag[i][j]` indica al descodificador de vídeo 30 que los elementos sintácticos `max_vps_num_reorder_pics[i][k][j]` y `max_vps_latency_increase_plus1[i][k][j]` se señalizan (por ejemplo, mediante el codificador de vídeo 20) para k en el intervalo de 0 a $(\text{NumLayersInIdList}[\text{output_layer_set_idx_minus1}[i+1]-1])$, inclusive.

40 **[0149]** El elemento sintáctico `max_vps_num_reorder_pics[i][k][j]` indica (por ejemplo, al descodificador de vídeo 30) el número máximo de imágenes permitido con `nuh_layer_id` igual a `layer_id_in_nuh[k]` que puede preceder a cualquier imagen con un identificador de capa `nuh` igual a `layer_id_in_nuh[k]` en el i -ésimo conjunto de capas de salida en el CVS en el orden de descodificación e ir detrás de esa imagen con un id de capa `nuh` igual a `layer_id_in_nuh[k]` en el orden de salida, y cuando `HighestTid` es igual a j . Cuando `max_vps_num_reorder_pics[i][k][j]` no está presente para j en el intervalo de 1 a $(\text{vps_max_sub_layers_minus1} - 1)$, inclusive, debido a que `sub_layer_dpb_info_present_flag[i][j]` es igual a 0, se infiere (por ejemplo, mediante el descodificador de vídeo 30) para que sea igual a `max_vps_num_reorder_pics[i][k][j-1]`. Por el contrario, en ejemplos donde `max_vps_num_reorder_pics[i][k][j]` no está presente para k en el intervalo de 0 a $(\text{NumLayersInIdList}[\text{output_layer_set_idx_minus1}[i+1]-2])$, inclusive, debido a que `sub_layer_dpb_info_present_flag[i][j]` es igual a 1 y `reorder_info_only_highest_layer_flag[i][j]` es igual a 0, el descodificador de vídeo 30 puede inferir `max_vps_num_reorder_pics[i][k][j]` para que sea igual a `max_vps_num_reorder_pics[i][NumLayersInIdList[output_layer_set_idx_minus1[i] + 1]-1][j-1]`.

55 **[0150]** En ejemplos donde `max_vps_latency_increase_plus1[i][k][j]` no es igual a 0, el descodificador de vídeo 30 puede usar el `max_vps_latency_increase_plus1[i][k][j]` para calcular el valor de `VpsMaxLatencyPictures[i][k][j]`, que especifica el número máximo de imágenes con `nuh_layer_id` igual a `layer_id_in_nuh[k]` en el i -ésimo conjunto de capas de salida que puede preceder a cualquier imagen con un identificador de capa `nuh` igual a `layer_id_in_nuh[k]` en el CVS en el orden de salida e ir detrás de esa imagen con id de capa `nuh` igual a `layer_id_in_nuh[k]` en el orden de descodificación cuando `HighestTid` es igual a j . En los ejemplos que `max_vps_latency_increase_plus1[i][k][j]` no está presente para j en el intervalo de 1 a $(\text{vps_max_sub_layers_minus1}-1)$, inclusive, debido a que el `sub_layer_dpb_info_present_flag[i][j]` es igual a 0, el descodificador de vídeo 30 puede inferir el `max_vps_latency_increase_plus1[i][k][j]` para que sea igual a `max_vps_latency_increase_plus1[i][k][j-1]`. En ejemplos donde `max_vps_latency_increase_plus1[i][k][j]` no está presente para k en el intervalo de 0 a $(\text{NumLayersInIdList}[\text{output_layer_set_idx_minus1}[i+1]-2])$, inclusive, debido a que `sub_layer_dpb_info_present_flag[i][j]` es igual a 1 y `reorder_info_only_highest_layer_flag[i][j]` es igual a 0, el

descodificador de vídeo 30 puede inferir el $\text{max_vps_latency_increase_plus1}[i][k][j]$ para que sea igual a $\text{max_vps_latency_increase_plus1}[i][\text{NumLayersInList}[\text{output_layer_set_idx_minus } 1][i+1]-1][j]$.

5 **[0151]** En los casos en que $\text{max_vps_latency_increase_plus1}[i][k][j]$ no es igual a 0, el valor de $\text{VpsMaxLatencyPictures}[i][k][j]$ se especifica mediante una fórmula de la siguiente manera:

$$\text{VpsMaxLatencyPictures}[i][k][j] = \text{max_vps_num_reorder_pics}[i][k][j] + \text{max_vps_latency_increase_plus1}[i][k][j] - 1$$

10 **[0152]** En los casos en que $\text{max_vps_latency_increase_plus1}[i][k][j]$ es igual a 0, no se expresa ningún límite correspondiente. El valor de $\text{max_vps_latency_increase_plus1}[i][k][j]$, en dichos ejemplos, estará en el intervalo de 0 a $(2^{32} - 2)$, inclusive.

15 **[0153]** La FIG. 6 es un diagrama de bloques que ilustra un codificador de vídeo 20 de ejemplo que puede implementar las técnicas para la señalización de parámetros de sub-DPB descritas en la presente divulgación. El codificador de vídeo 20 puede configurarse para codificar datos de vídeo de acuerdo con cualquier técnica de codificación de vídeo, incluyendo HEVC y H.264/AVC, así como extensiones ajustables a escala, multivista y 3D de dichas normas. El ejemplo de la FIG. 6 se explicará con referencia a la HEVC. En este sentido, el bucle de codificación de vídeo que se muestra en la FIG. 6 se puede aplicar a cada capa de un proceso de codificación de vídeo ajustable a escala (es decir, capas base y capas de mejora), a cada vista de un proceso de codificación de vídeo multivista, o a las vistas de textura y profundidad de un proceso de codificación de vídeo en 3D.

20 **[0154]** El codificador de vídeo 20 puede realizar la intracodificación y la intercodificación de bloques de vídeo dentro de fragmentos de vídeo. La intracodificación se basa en la predicción espacial para reducir o eliminar la redundancia espacial de un vídeo dentro de una trama o imagen de vídeo dada. La intercodificación se basa en la predicción temporal para reducir o eliminar la redundancia temporal de un vídeo dentro de tramas o imágenes adyacentes de una secuencia de vídeo. El intramodo (modo I) se puede referir a cualquiera de varios modos de compresión espacial. Los intermodos, tales como la predicción unidireccional (modo P) o la bipredicción (modo B), pueden hacer referencia a cualquiera de varios modos de compresión temporal. Además, el codificador de vídeo 20 puede realizar predicciones entre vistas y/o predicciones entre capas entre diferentes vistas o capas, como se ha descrito anteriormente.

25 **[0155]** En el ejemplo de la FIG. 6, el codificador de vídeo 20 incluye una memoria de datos de vídeo 40, una unidad de procesamiento de predicción 41, un DPB 64, un sumador 50, una unidad de procesamiento de transformada 52, una unidad de cuantificación 54 y una unidad de codificación por entropía 56. La unidad de procesamiento de predicción 41 incluye la unidad de estimación de movimiento y disparidad 42, la unidad de compensación de movimiento y disparidad 44 y la unidad de procesamiento de intrapredicción 46. Para la reconstrucción de bloques de vídeo, el codificador de vídeo 20 incluye también una unidad de cuantificación inversa 58, una unidad de procesamiento de transformada inversa 60 y un sumador 62. También se puede incluir un filtro de eliminación de bloques (no mostrado en la FIG. 6) para filtrar los límites de bloque, para eliminar distorsiones de efecto pixelado del vídeo reconstruido. Si se desea, el filtro de eliminación de bloques filtrará típicamente la salida del sumador 62. También se pueden usar filtros de bucle adicionales (en el bucle o tras el bucle), además del filtro de eliminación de bloques.

30 **[0156]** La memoria de datos de vídeo 40 puede almacenar datos de vídeo a codificar mediante los componentes del codificador de vídeo 20. Los datos de vídeo almacenados en la memoria de datos de vídeo 40 se pueden obtener, por ejemplo, a partir de la fuente de vídeo 18. La memoria de imágenes de referencia 64 es un ejemplo de un búfer de imágenes de descodificación (DPB), que almacena datos de vídeo de referencia para su uso en la codificación de datos de vídeo por el codificador de vídeo 20 (por ejemplo, en los modos de intracodificación o intercodificación, también denominados modos de codificación con intrapredicción o interpredicción). La memoria de datos de vídeo 40 y el DPB 64 pueden estar formadas por cualquiera de entre varios dispositivos de memoria, tales como memoria dinámica de acceso aleatorio (DRAM), incluyendo DRAM síncrona (SDRAM), RAM magnetoresistiva (MRAM), RAM resistiva (RRAM) u otros tipos de dispositivos de memoria. La memoria de datos de vídeo 40 y el DPB 64 pueden ser proporcionadas por el mismo dispositivo de memoria o por dispositivos de memoria independientes. En diversos ejemplos, la memoria de datos de vídeo 40 puede estar en un chip con otros componentes del codificador de vídeo 20, o fuera del chip con respecto a esos componentes.

35 **[0157]** Como se muestra en la FIG. 6, el codificador de vídeo 20 recibe datos de vídeo y puede configurarse para dividir los datos en bloques de vídeo. Esta partición también puede incluir la partición en fragmentos, mosaicos u otras unidades mayores, así como la división en bloques de vídeo, por ejemplo, de acuerdo con una estructura de árbol cuaternario de LCU y CU. El codificador de vídeo 20 ilustra, en general, los componentes que codifican bloques de vídeo dentro de un fragmento de vídeo que se va a codificar. El fragmento se puede dividir en múltiples bloques de vídeo (y, posiblemente, en conjuntos de bloques de vídeo denominados mosaicos). La unidad de procesamiento de predicción 41 puede seleccionar uno entre una pluralidad de posibles modos de codificación, tal como uno entre una pluralidad de modos de intracodificación, o uno entre una pluralidad de modos de

intercodificación o modos de codificación entre vistas, para el bloque de vídeo actual basándose en resultados de error (por ejemplo, la tasa de codificación y el nivel de distorsión). La unidad de procesamiento de predicción 41 puede proporcionar el bloque intracodificado o intercodificado resultante al sumador 50 para generar datos de bloque residuales y al sumador 62 para reconstruir el bloque codificado para su uso como imagen de referencia.

[0158] La unidad de procesamiento de intrapredicción 46, dentro de la unidad de procesamiento de predicción 41, puede realizar la codificación con intrapredicción del bloque de vídeo actual con respecto a uno o más bloques vecinos en la misma trama o fragmento que el bloque actual que vaya a codificarse, para proporcionar compresión espacial. La unidad de estimación de movimiento y disparidad 42 y la unidad de compensación de movimiento y disparidad 44 de la unidad de procesamiento de predicción 41 realizan la codificación con interpredicción y/o la codificación entre vistas del bloque de vídeo actual en relación con uno o más bloques predictivos de una o más imágenes de referencia, capas de imágenes de referencia, y/o vistas de referencia para proporcionar predicción temporal o entre vistas.

[0159] La unidad de estimación de movimiento y disparidad 42 puede estar configurada para determinar el modo de interpredicción y/o el modo de predicción entre vistas para un fragmento de vídeo de acuerdo con un patrón predeterminado para una secuencia de vídeo. El patrón predeterminado puede designar fragmentos de vídeo en la secuencia como fragmentos P o fragmentos B. La unidad de estimación de movimiento y disparidad 42 y la unidad de compensación de movimiento y disparidad 44 pueden estar sumamente integradas, pero se ilustran por separado con fines conceptuales. La estimación de movimiento, llevada a cabo por la unidad de estimación de movimiento y disparidad 42, es el proceso de generar vectores de movimiento, que estiman el movimiento para los bloques de vídeo. Un vector de movimiento, por ejemplo, puede indicar el desplazamiento de una PU de un bloque de vídeo de una trama o imagen de vídeo actual en relación con un bloque predictivo de una imagen de referencia. La estimación de disparidad, realizada por la unidad de estimación de movimiento y disparidad 42, es el proceso de generación de vectores de disparidad, que puede usarse para predecir un bloque codificado actualmente a partir de un bloque en una vista diferente.

[0160] Un bloque predictivo es un bloque que se determina que coincide estrechamente con la PU del bloque de vídeo que se va a codificar en términos de diferencia de píxel, que se puede determinar mediante una suma de diferencia absoluta (SAD), una 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 fraccionarios de las imágenes de referencia almacenadas en el DPB 64. Por ejemplo, el codificador de vídeo 20 puede interpolar valores de posiciones de un cuarto de píxel, posiciones de un octavo de píxel u otras posiciones fraccionarias de píxel de la imagen de referencia. Por lo tanto, la unidad de estimación de movimiento 42 puede realizar una búsqueda de movimiento con respecto a las posiciones de píxel completo y las posiciones de píxel fraccionario, y proporcionar un vector de movimiento con una precisión de píxel fraccionario.

[0161] La unidad de estimación de movimiento y disparidad 42 calcula un vector de movimiento (para predicción con compensación de movimiento) y/o un vector de movimiento de disparidad (para predicción con compensación de disparidad) para una PU de un bloque de vídeo en un fragmento sometido a intercodificación o predicción entre vistas, comparando la posición de la PU con la posición de un bloque predictivo de una imagen de referencia. La imagen de referencia puede seleccionarse a partir de una primera lista de imágenes de referencia (Lista 0) o una segunda lista de imágenes de referencia (Lista 1), cada una de las cuales identifica una o más imágenes de referencia almacenadas en el DPB 64. Para la predicción entre vistas, la imagen de referencia está en una vista diferente. La unidad de estimación de movimiento y disparidad 42 envía el vector de movimiento y/o el vector de disparidad calculados a la unidad de codificación por entropía 56 y la unidad de compensación de movimiento 44.

[0162] La compensación de movimiento y/o la compensación de disparidad, realizadas por la unidad de compensación de movimiento y disparidad 44, pueden implicar obtener o generar el bloque predictivo basándose en el vector de movimiento determinado mediante estimación de movimiento y/o estimación de paridad, realizar posiblemente interpolaciones hasta la precisión de subpíxel. Tras recibir el vector de movimiento y/o la disparidad para la PU del bloque de vídeo actual, la unidad de compensación de movimiento y disparidad 44 puede localizar el bloque predictivo al que apunta el vector de movimiento y/o el vector de disparidad en una de las listas de imágenes de referencia. El codificador de vídeo 20 forma un bloque de vídeo residual restando los valores de píxel del bloque predictivo a los valores de píxel del bloque de vídeo actual que se está codificando, formando valores de diferencia de píxel. Los valores de diferencia de píxel forman datos residuales para el bloque, y pueden incluir componentes de diferencia tanto de luma como de croma. El sumador 50 representa el componente o los componentes que realizan esta operación de resta. La unidad de compensación de movimiento y disparidad 44 también puede generar elementos sintácticos asociados a los bloques de vídeo y al fragmento de vídeo para su uso por el descodificador de vídeo 30 en la descodificación de los bloques de vídeo del fragmento de vídeo.

[0163] La unidad de procesamiento de intrapredicción 46 puede realizar la intrapredicción de un bloque actual, de forma alternativa a la interpredicción llevada a cabo por la unidad de estimación de movimiento y disparidad 42 y la unidad de compensación de movimiento y disparidad 44, como se ha descrito anteriormente. En particular, la unidad de procesamiento de intrapredicción 46 puede determinar un modo de intrapredicción a usar para codificar un bloque actual. En algunos ejemplos, la unidad de procesamiento de intrapredicción 46 puede codificar un bloque

actual usando diversos modos de intrapredicción, por ejemplo, durante pasadas de codificación separadas, y la unidad de procesamiento de intrapredicción 46 (o una unidad de selección de modo, en algunos ejemplos) puede seleccionar un modo de intrapredicción adecuado a usar a partir de los modos sometidos a prueba. Por ejemplo, la unidad de procesamiento de intrapredicción 46 puede calcular valores de velocidad-distorsión usando un análisis de velocidad-distorsión para los diversos modos de intrapredicción sometidos a prueba, y seleccionar el modo de intrapredicción que tenga las mejores características de velocidad-distorsión entre los modos sometidos a prueba. El análisis de velocidad-distorsión determina, en general, una cantidad de distorsión (o error) entre un bloque codificado y un bloque original no codificado que se codificó para producir el bloque codificado, así como una velocidad de transmisión de bits (es decir, un número de bits) usada para producir el bloque codificado. La unidad de procesamiento de intrapredicción 46 puede calcular relaciones a partir de las distorsiones y velocidades para los diversos bloques codificados, para determinar qué modo de intrapredicción presenta el mejor valor de velocidad-distorsión para el bloque.

[0164] En cualquier caso, tras seleccionar un modo de intrapredicción para un bloque, la unidad de procesamiento de intrapredicción 46 puede proporcionar información que indica el 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 acuerdo con las técnicas de esta divulgación. El codificador de vídeo 20 puede incluir, en el flujo de bits transmitido, datos de configuración, que pueden incluir una pluralidad de tablas de índices del modo de intrapredicción y una pluralidad de tablas de índices del modo de intrapredicción modificadas (también denominadas tablas de correlación de palabras de código), definiciones de contextos de codificación para diversos bloques e indicaciones de un modo de intrapredicción más probable, una tabla de índices del modo de intrapredicción y una tabla de índices del modo de intrapredicción modificada, a usar para cada uno de los contextos.

[0165] Después de que la unidad de procesamiento de predicción 41 genere el bloque predictivo para el bloque de vídeo actual, ya sea mediante interpredicción o intrapredicción, el codificador de vídeo 20 forma un bloque de vídeo residual restando el bloque predictivo al bloque de vídeo actual. Los datos de vídeo residuales del bloque residual se pueden incluir en una o más TU y aplicarse a la unidad de procesamiento de transformada 52. La unidad de procesamiento de transformada 52 transforma los datos de vídeo residuales en coeficientes de transformada residuales usando una transformada, tal como una transformada discreta del coseno (DCT) o una transformada conceptualmente similar. La unidad de procesamiento de transformada 52 puede convertir los datos de vídeo residuales de un dominio de píxel a un dominio de transformada, tal como un dominio de la frecuencia.

[0166] La unidad de procesamiento de transformada 52 puede enviar los coeficientes de transformada resultantes a la unidad de cuantificación 54. La unidad de cuantificación 54 cuantifica los coeficientes de transformada para reducir más la tasa de bits. El proceso de cuantificación puede reducir la profundidad de bits asociada a algunos, o a la totalidad, de los coeficientes. El grado de cuantificación se puede modificar ajustando un parámetro de cuantificación. En algunos ejemplos, la unidad de cuantificación 54 puede realizar a continuación una exploración de la matriz que incluye los coeficientes de transformada cuantificados. De forma alternativa, la unidad de codificación por entropía 56 puede realizar la exploración.

[0167] Seguidamente a la cuantificación, la unidad de codificación por entropía 56 codifica por entropía los coeficientes de transformada cuantificados. Por ejemplo, la unidad de codificación por entropía 56 puede realizar una codificación de longitud variable adaptativa al contexto (CAVLC), una codificación aritmética binaria adaptativa al contexto (CABAC), una codificación aritmética binaria adaptativa al contexto basada en la sintaxis (SBAC), una codificación por entropía por división de intervalos de probabilidad (PIPE) u otra metodología o técnicas de codificación por entropía. Tras la codificación por entropía mediante la unidad de codificación por entropía 56, el flujo de bits codificado se puede transmitir al descodificador de vídeo 30, o archivarse para su posterior transmisión o recuperación mediante el descodificador de vídeo 30. La unidad de codificación por entropía 56 también puede codificar por entropía los vectores de movimiento y los otros elementos sintácticos para el fragmento de vídeo actual que se está codificando.

[0168] La unidad de cuantificación inversa 58 y la unidad de procesamiento de transformada inversa 60 aplican una cuantificación inversa y una transformada inversa, respectivamente, para reconstruir el bloque residual en el dominio de píxel para su posterior uso como bloque de referencia de una imagen de referencia. Esta parte del codificador de vídeo 20 a veces se denomina bucle de reconstrucción, y decodifica eficazmente un bloque de vídeo codificado para su uso como imagen de referencia en la interpredicción. Las imágenes reconstruidas y/o las imágenes de capa reconstruidas se almacenan en el DPB 64.

[0169] La unidad de compensación de movimiento y disparidad 44 puede calcular un bloque de referencia sumando 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 y disparidad 44 también puede aplicar uno o más filtros de interpolación al bloque residual reconstruido para calcular valores de píxeles fraccionarios, para su uso en la estimación de movimiento. El sumador 62 añade el bloque residual reconstruido al bloque de predicción compensado por movimiento producido por la unidad de compensación de movimiento 44 para producir un bloque de referencia para su almacenamiento en el DPB 64. La unidad de estimación de movimiento y disparidad 42 y la

unidad de compensación de movimiento y disparidad 44 pueden usar el bloque de referencia como un bloque de referencia para realizar la interpredicción de un bloque en una posterior imagen o trama de vídeo.

5 **[0170]** Como se ha analizado anteriormente con referencia a la FIG. 5, el DPB 64 puede dividirse en una pluralidad de sub-DPB, donde cada sub-DPB está configurado para almacenar imágenes para un tipo diferente de capa en un proceso de codificación de vídeo multicapa. Como se ha analizado anteriormente, y se analizará más detalladamente a continuación con referencia a la FIG. 8, el codificador de vídeo 20 puede configurarse para realizar las técnicas de señalización de parámetros de sub-DPB de esta divulgación.

10 **[0171]** La FIG. 7 es un diagrama de bloques que ilustra un decodificador de vídeo 30 de ejemplo que puede implementar las técnicas de señalización de parámetros de sub-DPB descritas en la presente divulgación. En el ejemplo de la FIG. 7, el decodificador de vídeo 30 incluye una memoria de datos de vídeo 69, una unidad de descodificación por entropía 80, una unidad de procesamiento de predicción 81, una unidad de cuantificación inversa 86, una unidad de procesamiento de transformada inversa 88, un sumador 90 y un DPB 92. La unidad de
15 procesamiento de predicción 81 incluye la unidad de compensación de movimiento y disparidad 82 y la unidad de procesamiento de intrapredicción 84. En algunos ejemplos, el decodificador de vídeo 30 realiza una pasada de descodificación, en general, recíproca a la pasada de codificación descrita con respecto al codificador de vídeo 20 de la FIG. 6.

20 **[0172]** La memoria de datos de vídeo 69 puede almacenar datos de vídeo, tales como un flujo de bits de vídeo codificado que los componentes del decodificador de vídeo 30 van a descodificar. Los datos de vídeo almacenados en la memoria de datos de vídeo 69 se pueden obtener, por ejemplo, a partir de un dispositivo de almacenamiento 34, a partir de una fuente de vídeo local, tal como una cámara, mediante comunicación de red cableada o inalámbrica de datos de vídeo o accediendo a medios físicos de almacenamiento de datos. La memoria
25 de datos de vídeo 69 puede formar un búfer de imágenes codificadas (CPB) que almacena datos de vídeo codificados a partir de un flujo de bits de vídeo codificado.

30 **[0173]** Durante el procedimiento de descodificación, el decodificador de vídeo 30 recibe un flujo de bits de vídeo codificado, que representa bloques de vídeo de un fragmento de vídeo codificado y elementos sintácticos asociados, desde el codificador de vídeo 20. La unidad de descodificación por entropía 80 del decodificador de vídeo 30 puede realizar la descodificación por entropía del flujo de bits para generar coeficientes cuantificados, vectores de movimiento, vectores de disparidad y otros elementos sintácticos. La unidad de descodificación por entropía 80 reenvía los vectores de movimiento y otros elementos sintácticos a la unidad de procesamiento de predicción 81. El decodificador de vídeo 30 puede recibir los elementos sintácticos a nivel de fragmento de vídeo
35 y/o a nivel de bloque de vídeo.

40 **[0174]** Cuando el fragmento de vídeo se codifica como un fragmento 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 fragmento de vídeo actual basándose en un modo de intrapredicción señalado y en datos de bloques previamente descodificados de la trama o imagen actual. Cuando la trama de vídeo se codifica como un fragmento intercodificado (es decir, B o P), o fragmento predictivo entre vistas, la unidad de compensación de movimiento y disparidad 82 de la unidad de procesamiento de predicción 81 produce bloques predictivos para un bloque de vídeo del fragmento de vídeo actual, basándose en los vectores de movimiento, vectores de disparidad y otros elementos sintácticos recibidos desde la unidad de descodificación por entropía 80. Los bloques predictivos se pueden generar a partir de una de las imágenes de referencia dentro de una de las listas de imágenes de referencia. El decodificador de vídeo 30 puede construir las listas de tramas de referencia, la Lista 0 y la Lista 1, usando técnicas de construcción por defecto basándose en las imágenes de referencia almacenadas en el DPB 92.

50 **[0175]** La unidad de compensación de movimiento y disparidad 82 determina la información de predicción para un bloque de vídeo del fragmento de vídeo actual, analizando sintácticamente los vectores de movimiento y otros elementos sintácticos, y usa la información de predicción para producir los bloques predictivos para el bloque de vídeo actual que está siendo descodificado. Por ejemplo, la unidad de compensación de movimiento y disparidad 82 utiliza algunos de los elementos sintácticos recibidos para determinar un modo de predicción (por ejemplo, intrapredicción o interpredicción) utilizado para codificar los bloques de vídeo del fragmento de vídeo, un tipo de fragmento de interpredicción o predicción entre vistas (por ejemplo, fragmento B o fragmento P), información de construcción para una o más de las listas de imágenes de referencia para el fragmento, vectores de movimiento y/o vectores de disparidad para cada bloque de vídeo intercodificado del fragmento, estado de interpredicción para cada bloque de vídeo intercodificado del fragmento y otra información para descodificar los bloques de vídeo del
55 fragmento de vídeo actual.

60 **[0176]** La unidad de compensación de movimiento y disparidad 82 también puede realizar la interpolación basándose en filtros de interpolación. La unidad de compensación de movimiento y disparidad 82 puede usar filtros de interpolación como los usados por el codificador de vídeo 20 durante la codificación de los bloques de vídeo, para calcular valores interpolados para píxeles fraccionarios de bloques de referencia. En este caso, la unidad de compensación de movimiento y disparidad 82 puede determinar los filtros de interpolación utilizados por el

codificador de vídeo 20 a partir de los elementos sintácticos recibidos, y utilizar los filtros de interpolación para generar bloques predictivos.

5 **[0177]** La unidad de cuantificación inversa 86 cuantifica inversamente, es decir, descuantifica, los coeficientes de transformada cuantificados proporcionados en el flujo de bits y descodificados por la unidad de descodificación por entropía 80. El proceso de cuantificación inversa puede incluir el uso de un parámetro de cuantificación calculado por el codificador de vídeo 20 para cada bloque de vídeo del fragmento de vídeo para determinar un grado de cuantificación y, asimismo, un grado de cuantificación inversa que se debería aplicar. La unidad de procesamiento de transformada inversa 88 aplica una transformada inversa, por ejemplo, una DCT inversa, una transformada entera inversa o un proceso de transformada inversa conceptualmente similar, a los coeficientes de transformada a fin de generar bloques residuales en el dominio del píxel.

15 **[0178]** Después de que la unidad de compensación de movimiento y disparidad 82 genera el bloque predictivo para el bloque de vídeo actual, basándose en los vectores de movimiento y/o vectores de disparidad y otros elementos sintácticos, el descodificador de vídeo 30 forma un bloque de vídeo descodificado sumando los bloques residuales procedentes de la unidad de procesamiento de transformada inversa 88 a los bloques predictivos correspondientes generados por la unidad de compensación de movimiento 82. El sumador 90 representa el componente o los componentes que realizan esta operación de suma. Si se desea, también se puede aplicar un filtro de eliminación de bloques para filtrar los bloques descodificados, a fin de eliminar distorsiones de efecto pixelado. También se pueden usar otros filtros de bucle (bien en el bucle de codificación o bien después del bucle de codificación) para suavizar las transiciones de píxeles o mejorar de otro modo la calidad del vídeo.

25 **[0179]** El vídeo descodificado en una trama o imagen dada son a continuación almacenados en el DPB 92, que almacena bloques de imágenes de referencia (o imágenes en capas en un proceso de codificación de vídeo multivista) usadas para la compensación de movimiento posterior. El DPB 92 almacena también vídeo decodificado para su presentación en un dispositivo de visualización, tal como el dispositivo de visualización 32 de la FIG. 1. Como se ha analizado anteriormente con referencia a la FIG. 5, el DPB 92 puede dividirse en una pluralidad de sub-DPB, donde cada sub-DPB está configurado para almacenar imágenes para un tipo diferente de capa en un proceso de codificación de vídeo multicapa. Como se ha analizado anteriormente, y se analizará más detalladamente a continuación con referencia a la FIG. 9, el descodificador de vídeo 300 puede configurarse para realizar las técnicas de señalización de parámetros de sub-DPB de esta divulgación.

35 **[0180]** La FIG. 8 es un diagrama de flujo que muestra un procedimiento de codificación de ejemplo de acuerdo con las técnicas de esta divulgación. Las técnicas de la FIG. 8 pueden implementarse mediante una o más unidades de hardware del codificador de vídeo 20, incluido el DPB 64.

40 **[0181]** En un ejemplo de la divulgación, el codificador de vídeo 20 puede configurarse para reconstruir una pluralidad de imágenes de un flujo de bits de vídeo codificado (800), y almacenar la pluralidad de imágenes en los uno o más sub-DPB (802). Es decir, el DPB 64 puede dividirse en uno o más sub-DBP. El codificador de vídeo 20 puede configurarse adicionalmente para generar un conjunto respectivo de parámetros de sub-DPB para cada punto de operación respectivo del flujo de bits de vídeo codificado (804), aplicar el conjunto respectivo de parámetros de sub-DPB a todas las capas de un conjunto de capas de salida para cada punto de operación respectivo (806), y realizar un proceso de gestión de sub-DPB en los uno o más sub-DPB de acuerdo con los respectivos conjuntos individuales de parámetros de sub-DPB generados (808). En un ejemplo de la divulgación, los parámetros de sub-DPB incluyen al menos uno de entre un elemento sintáctico indicativo de un número máximo de imágenes reordenadas permitidas para los uno o más sub-DPB y un elemento sintáctico indicativo de una latencia máxima permitida para los uno o más sub-DPB. El codificador de vídeo 20 puede configurarse adicionalmente para señalar el conjunto respectivo de parámetros de sub-DPB en el flujo de bits de vídeo codificado (810).

50 **[0182]** En otro ejemplo de la divulgación, el codificador de vídeo 20 puede configurarse para generar el conjunto respectivo de parámetros de sub-DPB en un conjunto de parámetros de vídeo (VPS). En otro ejemplo de la divulgación, el codificador de vídeo 20 puede configurarse para generar un primer elemento sintáctico de subcapa respectivo que indica si los parámetros de sub-DPB se señalizan o no para cada subcapa de cada conjunto de capas de salida respectivo. En otro ejemplo de la divulgación, el codificador de vídeo 20 puede configurarse para generar, en el caso de que el primer elemento sintáctico de subcapa indique que los parámetros de sub-DPB pueden señalizarse para cada subcapa de un conjunto de capas de salida particular, un segundo elemento sintáctico de subcapa respectivo que indica si los parámetros de sub-DPB se señalizan o no para las subcapas respectivas del conjunto de capas de salida particular. En otro ejemplo de la divulgación, el codificador de vídeo 20 puede configurarse para generar, en el caso de que el segundo elemento sintáctico de subcapa respectivo indique que los parámetros de sub-DPB se señalizan para la subcapa particular, un conjunto adicional de parámetros de sub-DPB para la subcapa particular.

65 **[0183]** En otro ejemplo de la divulgación, el proceso de gestión del sub-DPB incluye al menos uno de entre el marcado de imágenes como no utilizadas de referencia, marcado de imágenes para salida y eliminación de imágenes del sub-DPB, y en el que la realización del proceso de gestión de sub-DPB comprende realizar el proceso

de gestión de sub-DPB en un nivel de unidad de acceso utilizando los conjuntos respectivos de parámetros de sub-DPB recibidos. El codificador de vídeo 20 puede configurarse para realizar el proceso de gestión de sub-DPB en el nivel de la unidad de acceso produciendo imágenes basadas en el número de unidades de acceso en un DPB que contienen al menos una imagen marcada como necesaria para la salida, marcando las imágenes como no utilizadas de referencia en un nivel por capa (por ejemplo, para cada capa de forma independiente) y eliminando imágenes del sub-DPB en un nivel por capa (por ejemplo, para cada capa de forma independiente).

[0184] La FIG. 9 es un diagrama de flujo que muestra un procedimiento de descodificación de ejemplo de acuerdo con las técnicas de la presente divulgación. Las técnicas de la FIG. 9 pueden implementarse mediante una o más unidades de hardware del descodificador de vídeo 30, incluido el DPB 92.

[0185] En un ejemplo de la divulgación, el descodificador de vídeo 30 puede configurarse para recibir un flujo de bits de vídeo codificado que incluye una pluralidad de imágenes (900) y almacenar la pluralidad de imágenes en los uno o más sub-DPB (902). Es decir, el DPB 92 puede dividirse en uno o más sub-DBP. El descodificador de vídeo 30 puede configurarse adicionalmente para recibir un conjunto respectivo de parámetros de sub-DPB para cada punto de operación respectivo del flujo de bits de vídeo codificado (904), aplicar el conjunto respectivo de parámetros de sub-DPB a todas las capas de un conjunto de capas de salida para cada punto de operación respectivo (906), y realizar un proceso de gestión de sub-DPB en los uno o más sub-DPB de acuerdo con los respectivos conjuntos individuales de parámetros de sub-DPB recibidos (908). En un ejemplo de la divulgación, los parámetros de sub-DPB incluyen al menos uno de entre un elemento sintáctico indicativo de un número máximo de imágenes reordenadas permitidas para los uno o más sub-DPB y un elemento sintáctico indicativo de una latencia máxima permitida para los uno o más sub-DPB.

[0186] En otro ejemplo de la divulgación, el descodificador de vídeo 30 puede configurarse para recibir el conjunto respectivo de parámetros de sub-DPB en un conjunto de parámetros de vídeo (VPS). En otro ejemplo de la divulgación, el descodificador de vídeo 30 puede configurarse para recibir un primer elemento sintáctico de subcapa respectivo que indica si los parámetros de sub-DPB se señalizan o no para cada subcapa de cada conjunto de capas de salida respectivo.

[0187] En otro ejemplo de la divulgación, el descodificador de vídeo 30 puede configurarse para aplicar, en el caso de que el primer elemento sintáctico de subcapa indique que los parámetros de sub-DPB no están señalizados para cada subcapa de un conjunto de capas de salida particular, el conjunto de parámetros de sub-DPB respectivo a todas las subcapas del conjunto de capas de salida particular. En otro ejemplo de la divulgación, el descodificador de vídeo 30 puede configurarse para recibir, en el caso de que el primer elemento sintáctico de subcapa indique que los parámetros de sub-DPB pueden señalizarse para cada subcapa de un conjunto de capas de salida particular, un segundo elemento sintáctico de subcapa respectivo que indica si los parámetros de sub-DPB se señalizan o no para las subcapas respectivas del conjunto de capas de salida particular.

[0188] En otro ejemplo de la divulgación, el descodificador de vídeo 30 puede configurarse para aplicar, en el caso de que el segundo elemento sintáctico de subcapa respectivo indique que los parámetros de sub-DPB no están señalizados para una subcapa particular, el conjunto de parámetros de sub-PDB respectivo para la subcapa particular. Además, en el caso de que el segundo elemento sintáctico de subcapa respectivo indique que los parámetros de sub-DPB están señalizados para la subcapa particular, el descodificador de vídeo 30 puede configurarse para recibir un conjunto adicional de parámetros de sub-DPB para la subcapa particular.

[0189] En otro ejemplo de la divulgación, el descodificador de vídeo 30 puede configurarse para aplicar, en el caso de que el segundo elemento sintáctico de subcapa respectivo indique que los parámetros de sub-DPB no están señalizados para una subcapa particular con un ID de subcapa X, un último conjunto recibido respectivo de parámetros de sub-DPB para las subcapas que tienen un ID de subcapa mayor que X.

[0190] En otro ejemplo de la divulgación, el proceso de gestión del sub-DPB incluye al menos uno de entre el marcado de imágenes como no utilizadas de referencia, marcado de imágenes para salida y eliminación de imágenes del sub-DPB, y en el que la realización del proceso de gestión de sub-DPB comprende realizar el proceso de gestión de sub-DPB en un nivel de unidad de acceso utilizando los conjuntos respectivos de parámetros de sub-DPB recibidos. El descodificador de vídeo 30 puede configurarse para realizar el proceso de gestión de sub-DPB en el nivel de la unidad de acceso produciendo imágenes basadas en el número de unidades de acceso en un DPB que contienen al menos una imagen marcada como necesaria para la salida, marcando las imágenes como no utilizadas de referencia en un nivel por capa (por ejemplo, para cada capa de forma independiente) y eliminando imágenes del sub-DPB en un nivel por capa (por ejemplo, para cada capa de forma independiente).

[0191] Se debe reconocer que, dependiendo del ejemplo, determinadas acciones o eventos de cualquiera de las técnicas descritas en el presente documento se pueden realizar en una secuencia distinta, se pueden añadir, fusionar o excluir por completo (por ejemplo, no todas las acciones o eventos descritos son necesarios para la puesta en práctica de las técnicas). Por otro lado, en determinados ejemplos, las acciones o eventos pueden tener lugar simultáneamente, por ejemplo, a través de procesamiento de múltiples subprocesos, procesamiento de interrupciones o múltiples procesadores, en lugar de secuencialmente.

[0192] En uno o más ejemplos, las funciones descritas se pueden implementar en hardware, software, firmware o cualquier combinación de los mismos. Si se implementan en software, las funciones se pueden almacenar en, o transmitir por, un medio legible por ordenador, como una o más instrucciones o código, y ejecutar mediante una unidad de procesamiento basada en hardware. Los medios legibles por ordenador pueden incluir medios de almacenamiento legibles por ordenador, que corresponden a un medio tangible tal como unos medios de almacenamiento de datos, o medios de comunicación que incluyen cualquier medio que facilita la transferencia de un programa informático de un lugar a otro, por ejemplo, de acuerdo con un protocolo de comunicación. De esta manera, los medios legibles por ordenador pueden corresponder, en general, a (1) medios de almacenamiento tangibles legibles por ordenador que son no transitorios o (2) un medio de comunicación tal como una señal o una onda portadora. Los medios de almacenamiento de datos pueden ser cualquier medio disponible al que se pueda acceder desde uno o más ordenadores o uno o más procesadores para recuperar instrucciones, código y/o estructuras de datos para la implementación de las técnicas descritas en la presente divulgación. Un producto de programa informático puede incluir un medio legible por ordenador. Como se usa en el presente documento, el término "señalización" puede incluir almacenar o incluir de otro modo datos con un flujo de bits codificado. En otras palabras, en diversos ejemplos de acuerdo con esta divulgación, el término 'señalización' puede estar asociado con la comunicación de datos en tiempo real, o de forma alternativa, la comunicación que no se realiza en tiempo real.

[0193] A modo de ejemplo, y no de limitación, dichos medios de almacenamiento legibles por ordenador pueden comprender RAM, ROM, EEPROM, CD-ROM u otro almacenamiento en disco óptico, almacenamiento en disco magnético u otros dispositivos de almacenamiento magnético, memoria flash o cualquier otro medio que se pueda usar para almacenar código de programa deseado en forma de instrucciones o estructuras de datos y al que se pueda acceder mediante un ordenador. Además, cualquier conexión recibe apropiadamente la denominación de medio legible por ordenador. Por ejemplo, si las instrucciones se transmiten desde un sitio web, un servidor u otro origen remoto usando un cable coaxial, un cable de fibra óptica, un par trenzado, una línea de abonado digital (DSL) o unas tecnologías inalámbricas tales como infrarrojos, radio y microondas, entonces el cable coaxial, el cable de fibra óptica, el par trenzado, la DSL o las tecnologías inalámbricas tales como infrarrojos, radio y microondas están incluidos en la definición de medio. Sin embargo, se deberá entender que los medios de almacenamiento legibles por ordenador y los medios de almacenamiento de datos no incluyen conexiones, ondas portadoras, señales u otros medios transitorios, sino que, en cambio, están dirigidos a medios de almacenamiento no transitorio tangibles. El término disco, como se usa en el presente documento, incluye el disco compacto (CD), disco láser, disco óptico, disco versátil digital (DVD), disco flexible y disco Blu-ray, donde algunos discos normalmente reproducen datos magnéticamente, mientras que otros discos reproducen datos ópticamente con láseres. Las combinaciones de los anteriores también se deben incluir dentro del alcance de los medios legibles por ordenador.

[0194] Uno o más procesadores, tales como uno o más procesadores de señales digitales (DSP), microprocesadores de propósito general, circuitos integrados específicos de la aplicación (ASIC), matrices lógicas programables in situ (FPGA) u otros circuitos lógicos integrados o discretos equivalentes pueden ejecutar las instrucciones. Por consiguiente, el término "procesador", como se usa en el presente documento, se puede referir a cualquier estructura anterior o a cualquier otra estructura adecuada para la implementación de las técnicas descritas en el presente documento. Además, en algunos aspectos, la funcionalidad descrita en el presente documento se puede proporcionar en módulos de hardware y/o de software dedicados configurados para la codificación y la decodificación, o incorporarse en un códec combinado. Además, las técnicas se podrían implementar por completo en uno o más circuitos o elementos lógicos.

[0195] Las técnicas de esta divulgación se pueden implementar en una amplia variedad de dispositivos o aparatos, incluyendo un teléfono inalámbrico, un circuito integrado (IC) o un conjunto de IC (por ejemplo, un conjunto de chips). En esta divulgación se describen diversos componentes, módulos o unidades para destacar aspectos funcionales de dispositivos configurados para realizar las técnicas divulgadas, pero no se requiere necesariamente su realización por diferentes unidades de hardware. En su lugar, como se ha descrito anteriormente, diversas unidades se pueden combinar en una unidad de hardware de códec o proporcionar mediante un grupo de unidades de hardware interoperativas, incluidos uno o más procesadores como los descritos anteriormente, junto con software y/o firmware adecuados.

REIVINDICACIONES

1. Un procedimiento de descodificación de datos de vídeo que usa un búfer de imágenes descodificadas - DPB - dividido en uno o más sub-DPB, el procedimiento que comprende:

5 recibir (900), a partir de un flujo de bits de vídeo codificado, una pluralidad de conjuntos de capas de salida, en el que cada conjunto de capas de salida de la pluralidad de conjuntos de capas de salida incluye una pluralidad respectiva de subcapas que son capas temporales ajustables a escala de un flujo de bits temporal ajustable a escala, en el que cada respectiva pluralidad de subcapas incluye una primera subcapa y una o
10 más subcapas distintas de la primera subcapa;

descodificar cada pluralidad respectiva de subcapas de la pluralidad de conjuntos de capas de salida;

15 almacenar (902) la pluralidad de cada pluralidad de subcapas descodificada respectiva en uno o más sub-DPB;

recibir (904), a partir del flujo de bits de vídeo codificado, un primer elemento sintáctico respectivo para cada conjunto de capas de salida de la pluralidad de conjuntos de capas de salida, en el que cada primer elemento sintáctico respectivo es indicativo de si los parámetros de sub-DPB se señalizan para más de una subcapa del conjunto de capas de salida respectivo o no;

20 condicional a que el primer elemento sintáctico respectivo recibido para un conjunto de capas de salida particular sea indicativo de que los parámetros de sub-DPB se señalizan para más de una subcapa del conjunto de capas de salida particular, recibir, a partir del flujo de bits de vídeo codificado, un segundo elemento sintáctico respectivo para cada subcapa de las una o más subcapas distintas de la primera subcapa del conjunto de capas de salida particular, en el que cada segundo elemento sintáctico respectivo es indicativo de si parámetros de sub-DPB se señalizan para la subcapa respectiva de las una o más subcapas distintas de la primera subcapa del conjunto de capas de salida particular o no;

30 recibir, a partir del flujo de bits de vídeo codificado, un conjunto respectivo de parámetros de sub-DPB para la primera subcapa de cada pluralidad respectiva de subcapas del conjunto de capas de salida particular;

35 para cada subcapa respectiva de las una o más subcapas distintas de la primera subcapa de un conjunto de capas de salida, dependiendo del valor del segundo elemento sintáctico respectivo recibido para la subcapa, o inferir un conjunto respectivo de parámetros de sub-DPB a partir de los parámetros de sub-DPB para una subcapa anterior o recibir un conjunto respectivo de parámetros de sub-DPB del flujo de bits de vídeo codificado; y

40 realizar (908) un proceso de gestión de sub-DPB en los uno o más sub-DPB de acuerdo con cada conjunto respectivo de parámetros de sub-DPB recibidos.

2. El procedimiento de la reivindicación 1, en el que cada conjunto respectivo de parámetros de sub-DPB para la primera subcapa de cada pluralidad respectiva de subcapas incluye:

45 un tercer elemento sintáctico respectivo indicativo de un número máximo de imágenes reordenadas permitidas para los uno o más sub-DPB; y

50 un cuarto elemento sintáctico respectivo indicativo de una latencia máxima permitida para los uno o más sub-DPB.

3. El procedimiento de la reivindicación 1, en el que recibir el conjunto respectivo de parámetros de sub-DPB comprende recibir el conjunto respectivo de parámetros de sub-DPB en un conjunto de parámetros de vídeo - VPS.

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

55 aplicar, basado en el primer elemento sintáctico respectivo que indica que los parámetros de sub-DPB no están señalizados para cada subcapa de un segundo conjunto de capas de salida particular de la pluralidad de conjuntos de capas de salida, el conjunto respectivo de parámetros de sub-DPB del segundo conjunto de capas de salida particular a todas las subcapas del segundo conjunto de capas de salida particular.

60 **5.** El procedimiento de la reivindicación 1, en el que el proceso de gestión de sub-DPB incluye al menos uno de marcar imágenes como no utilizadas para referencia, marcar imágenes para salida, o eliminar imágenes del sub-DPB, y en el que realizarel proceso de gestión de sub-DPB comprende realizar el proceso de gestión de sub-DPB en un nivel de unidad de acceso utilizando los conjuntos respectivos de parámetros de sub-DPB recibidos.

65

6. El procedimiento de la reivindicación 5, en el que realizar el proceso de gestión de sub-DPB en el nivel de la unidad de acceso comprende:

- 5 producir imágenes basándose en el número de unidades de acceso en un DPB que contienen al menos una imagen marcada como necesaria para la salida;
- marcar imágenes como no utilizadas para referencia en un nivel por capa; y
- 10 eliminar imágenes del sub-DPB en un nivel por capa.

7. Un procedimiento de codificación de datos de vídeo que usa un búfer de imágenes descodificadas - DPB - dividido en uno o más sub-DPB, el procedimiento que comprende:

- 15 reconstruir (800) una pluralidad de conjuntos de capas de salida de un flujo de bits de vídeo codificado, en el que cada conjunto de capas de salida de la pluralidad de conjuntos de capas de salida incluye una pluralidad respectiva de subcapas que son capas temporales ajustables a escala de un flujo de bits temporal ajustable a escala, en el que cada pluralidad respectiva de subcapas incluye una primera subcapa y una o más subcapas distintas de la primera subcapa, en el que reconstruir la pluralidad de conjuntos de capas de salida incluye reconstruir cada pluralidad respectiva de subcapas de la pluralidad de conjuntos de capas de salida;

20 almacenar (802) cada pluralidad respectiva reconstruida de subcapas en los uno o más sub-DPB;

- 25 generar (804) un primer elemento sintáctico respectivo para cada conjunto de capas de salida de la pluralidad de conjuntos de capas de salida, en el que cada primer elemento sintáctico respectivo es indicativo de si parámetros de sub-DPB se señalizan para más de una subcapa del conjunto de capas de salida respectivo o no;

30 condicional a que el primer elemento sintáctico respectivo generado para un conjunto de capas de salida particular sea indicativo de que parámetros de sub-DPB se señalizan para más de una subcapa del conjunto de capas de salida particular, generar un segundo elemento sintáctico respectivo para cada subcapa de las una o más subcapas distintas de la primera subcapa del conjunto de capas de salida particular, en el que cada segundo elemento sintáctico respectivo es indicativo de si parámetros de sub-DPB se señalizan para la subcapa respectiva de las una o más subcapas distintas de la primera subcapa del conjunto de capas de salida particular o no;

35 generar un conjunto respectivo de parámetros de sub-DPB para la primera subcapa de cada pluralidad respectiva de subcapas y para cada subcapa respectiva de las una o más subcapas distintas de la primera subcapa de cada pluralidad respectiva de subcapas;

40 realizar (808) un proceso de gestión de sub-DPB en los uno o más sub-DPB de acuerdo con cada conjunto respectivo de parámetros de sub-DPB generados;

45 generar (810), para la primera subcapa de cada pluralidad respectiva de subcapas, cada conjunto respectivo de parámetros de sub-DPB en el flujo de bits de vídeo codificado;

50 para cada subcapa respectiva de las una o más subcapas distintas de la primera subcapa de un conjunto de capas de salida, dependiendo del valor del segundo elemento sintáctico respectivo generado para la subcapa, o generar un conjunto respectivo de parámetros de sub-DPB en el flujo de bits de vídeo codificado o no generar un conjunto respectivo de parámetros de sub-DPB en el flujo de bits de vídeo codificado.

8. Un aparato configurado para descodificar datos de vídeo que usa un búfer de imágenes descodificadas - DPB - dividido en una pluralidad de sub-DPB, el aparato que comprende:

55 uno o más dispositivos de memoria que comprenden uno o más sub-DPB configurados para almacenar datos de vídeo; y

un descodificador de vídeo configurado para:

60 recibir, a partir de un flujo de bits de vídeo codificado, una pluralidad de conjuntos de capas de salida, en el que cada conjunto de capas de salida de la pluralidad de conjuntos de capas de salida incluye una pluralidad respectiva de subcapas que son capas temporales ajustables a escala de un flujo de bits temporal ajustable a escala, en el que cada respectiva pluralidad de subcapas incluye una primera subcapa y una o más subcapas distintas de la primera subcapa;

65 descodificar cada pluralidad respectiva de subcapas de la pluralidad de conjuntos de capas de salida;

almacenar cada pluralidad de subcapas descodificada respectiva en los uno o más sub-DPB;

5 recibir, a partir del flujo de bits de vídeo codificado, un primer elemento sintáctico respectivo para cada conjunto de capas de salida de la pluralidad de conjuntos de capas de salida, en el que cada primer elemento sintáctico respectivo es indicativo de si los parámetros de sub-DPB se señalizan para más de una subcapa del conjunto de capas de salida respectivo o no;

10 condicional a que el primer elemento sintáctico respectivo recibido para un conjunto de capas de salida particular sea indicativo de que parámetros de sub-DPB se señalizan para más de una subcapa del conjunto de capas de salida particular, recibir, a partir del flujo de bits de vídeo codificado, un segundo elemento sintáctico respectivo para cada subcapa de las una o más subcapas distintas de la primera subcapa del conjunto de capas de salida particular, en el que cada segundo elemento sintáctico respectivo es indicativo de si parámetros de sub-DPB se señalizan para la subcapa respectiva de las una o más subcapas distintas de la primera subcapa del conjunto de capas de salida particular o no;

15 recibir, a partir del flujo de bits codificado, un conjunto respectivo de parámetros de sub-DPB para la primera subcapa de cada pluralidad respectiva de subcapas;

20 para cada subcapa respectiva de las una o más subcapas distintas de la primera subcapa de un conjunto de capas de salida, dependiendo del valor del segundo elemento sintáctico respectivo recibido para la subcapa, o inferir un conjunto respectivo de parámetros de sub-DPB a partir de los parámetros de sub-DPB para una subcapa anterior o recibir un conjunto respectivo de parámetros de sub-DPB del flujo de bits de vídeo codificado; y

25 realizar un proceso de gestión de sub-DPB en los uno o más sub-DPB de acuerdo con cada conjunto respectivo de parámetros de sub-DPB recibidos.

30 **9.** El aparato de la reivindicación 8, en el que cada conjunto respectivo de parámetros de sub-DPB para la primera subcapa de cada pluralidad respectiva de subcapas incluye:

un tercer elemento sintáctico respectivo indicativo de un número máximo de imágenes reordenadas permitidas para los uno o más sub-DPB; y

35 un cuarto elemento sintáctico respectivo indicativo de una latencia máxima permitida para los uno o más sub-DPB.

10. El aparato de la reivindicación 8, en el que el descodificador de vídeo está configurado para recibir el conjunto respectivo de parámetros de sub-DPB en un conjunto de parámetros de vídeo - VPS.

40 **11.** El aparato de la reivindicación 8, en el que el descodificador de vídeo está configurado además para:

45 aplicar, en base al primer elemento sintáctico respectivo que indica que los parámetros de sub-DPB no están señalizados para cada subcapa de un segundo conjunto de capas de salida particular, el conjunto respectivo de parámetros de sub-DPB del segundo conjunto de capas de salida particular a todas las subcapas del segundo conjunto de capas de salida particular.

50 **12.** El aparato de la reivindicación 8, en el que el proceso de gestión de sub-DPB incluye al menos uno de marcar imágenes como no utilizadas para referencia, marcar imágenes para salida, o eliminar imágenes del sub-DPB, y en el que realizarel proceso de gestión de sub-DPB comprende realizar el proceso de gestión de sub-DPB en un nivel de unidad de acceso utilizando los conjuntos respectivos de parámetros de sub-DPB recibidos.

13. El aparato de acuerdo con la reivindicación 12, en el que el descodificador de vídeo está configurado, además, para:

55 producir imágenes basándose en el número de unidades de acceso en un DPB que contienen al menos una imagen marcada como necesaria para la salida;

marcar imágenes como no utilizadas para referencia en un nivel por capa; y

60 eliminar imágenes del sub-DPB en un nivel por capa.

14. Un aparato configurado para codificar datos de vídeo que usa un búfer de imágenes descodificadas - DPB - dividido en una pluralidad de sub-DPB, el aparato que comprende:

65 uno o más dispositivos de memoria que comprenden los uno o más sub-DPB configurados para almacenar datos de vídeo; y

un codificador de vídeo configurado para:

5 reconstruir una pluralidad de conjuntos de capas de salida de un flujo de bits de vídeo codificado, en el que cada conjunto de capas de salida de la pluralidad de conjuntos de capas de salida incluye una pluralidad respectiva de subcapas que son capas temporales ajustables a escala de un flujo de bits temporal ajustable a escala, en el que cada pluralidad respectiva de subcapas incluye una primera subcapa y una o más subcapas distintas de la primera subcapa, en el que reconstruir la pluralidad de conjuntos de capas de salida incluye reconstruir cada pluralidad respectiva de subcapas de la pluralidad de conjuntos de capas de salida;

almacenar cada pluralidad respectiva reconstruida de subcapas en los uno o más sub-DPB;

15 generar un primer elemento sintáctico respectivo para cada conjunto de capas de salida de la pluralidad de conjuntos de capas de salida, en el que cada primer elemento sintáctico respectivo es indicativo de si los parámetros de sub-DPB se señalizan para más de una subcapa del conjunto de capas de salida respectivo o no;

20 condicional a que el primer elemento sintáctico respectivo generado para un conjunto de capas de salida particular sea indicativo de que parámetros de sub-DPB se señalizan para más de una subcapa del conjunto de capas de salida particular, generar un segundo elemento sintáctico respectivo para cada subcapa de las una o más subcapas distintas de la primera subcapa del conjunto de capas de salida particular, en el que cada segundo elemento sintáctico respectivo es indicativo de si los parámetros de sub-DPB se señalizan para la subcapa respectiva de las una o más subcapas distintas de la primera subcapa del conjunto de capas de salida particular o no;

25 generar un conjunto respectivo de parámetros de sub-DPB para la primera subcapa de cada pluralidad respectiva de subcapas y para cada subcapa respectiva de las una o más subcapas distintas de la primera subcapa de cada pluralidad respectiva de subcapas;

30 realizar un proceso de gestión de sub-DPB en los uno o más sub-DPB de acuerdo con cada conjunto respectivo de parámetros de sub-DPB generados;

35 generar, para la primera subcapa de cada pluralidad respectiva de subcapas, cada conjunto respectivo de parámetros de sub-DPB en el flujo de bits de vídeo codificado; y

40 para cada subcapa respectiva de las una o más subcapas distintas de la primera subcapa de un conjunto de capas de salida, dependiendo del valor del segundo elemento sintáctico respectivo generado para la subcapa, o generar un conjunto respectivo de parámetros de sub-DPB en el flujo de bits de vídeo codificado o no generar un conjunto respectivo de parámetros de sub-DPB en el flujo de bits de vídeo codificado.

45 **15.** Un medio de almacenamiento legible por ordenador que tiene instrucciones almacenadas que, cuando se ejecutan, hacen que uno o más procesadores de un dispositivo lleven a cabo un procedimiento de acuerdo con cualquiera de las reivindicaciones 1 a 7.

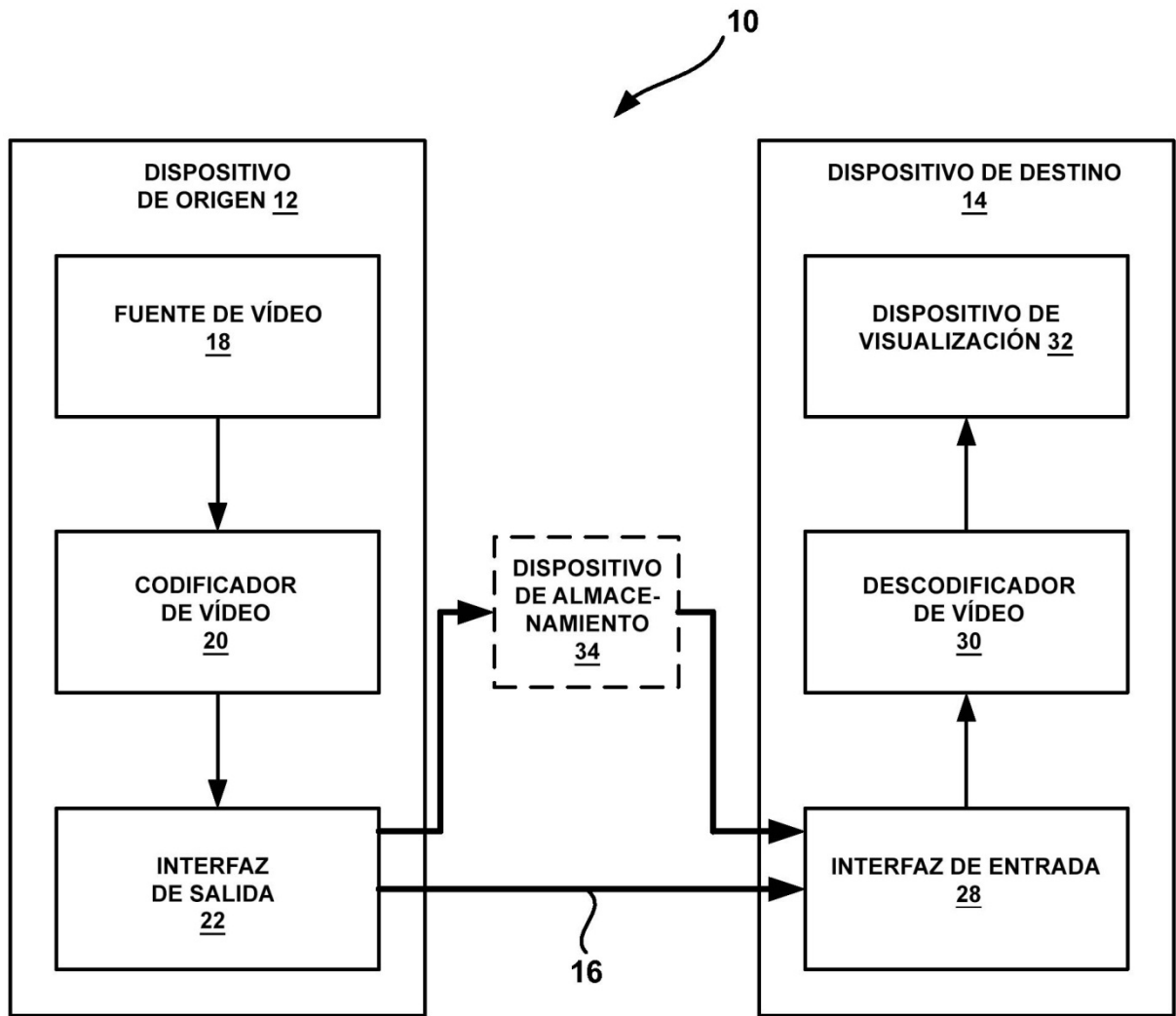


FIG. 1

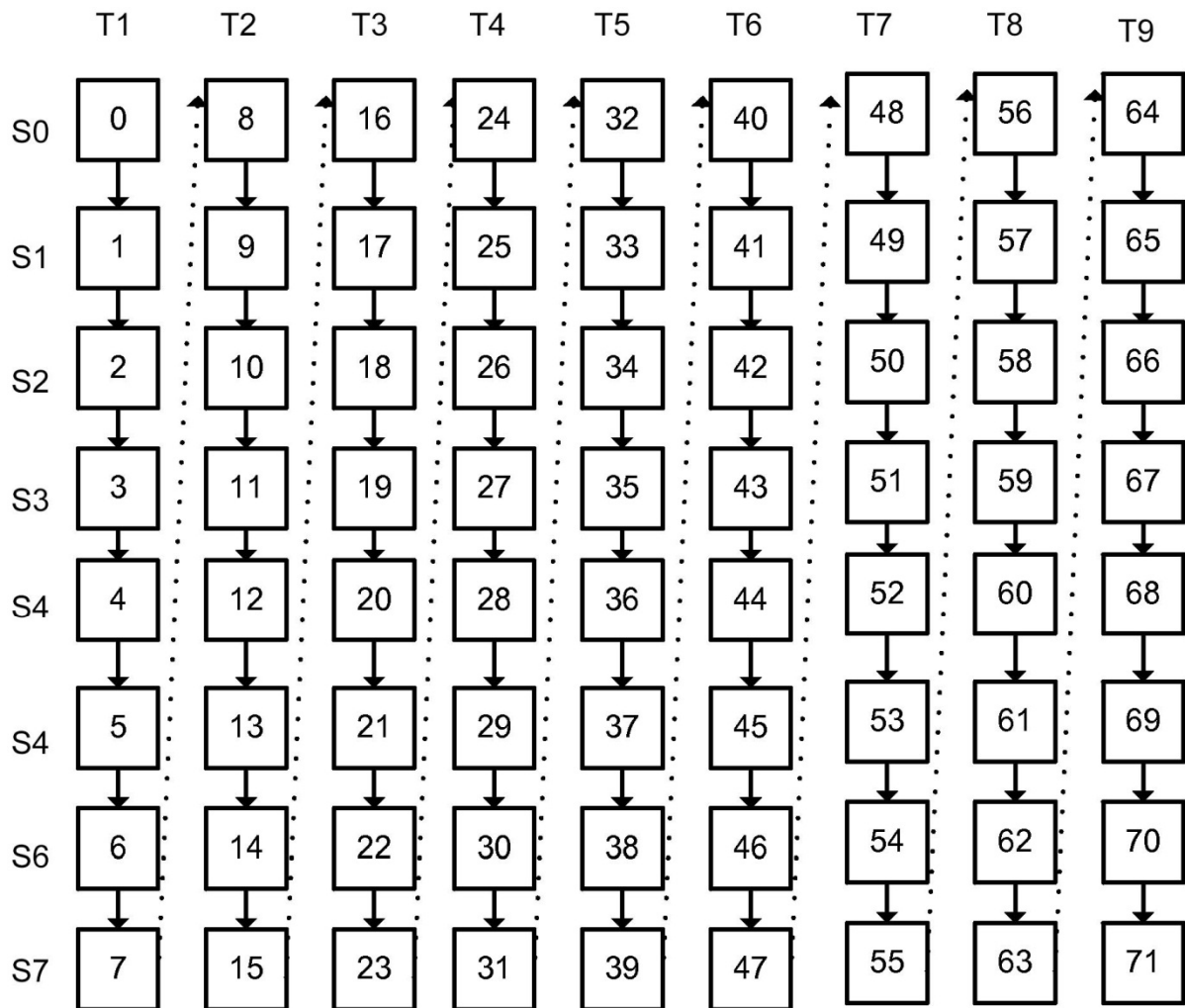


FIG. 2

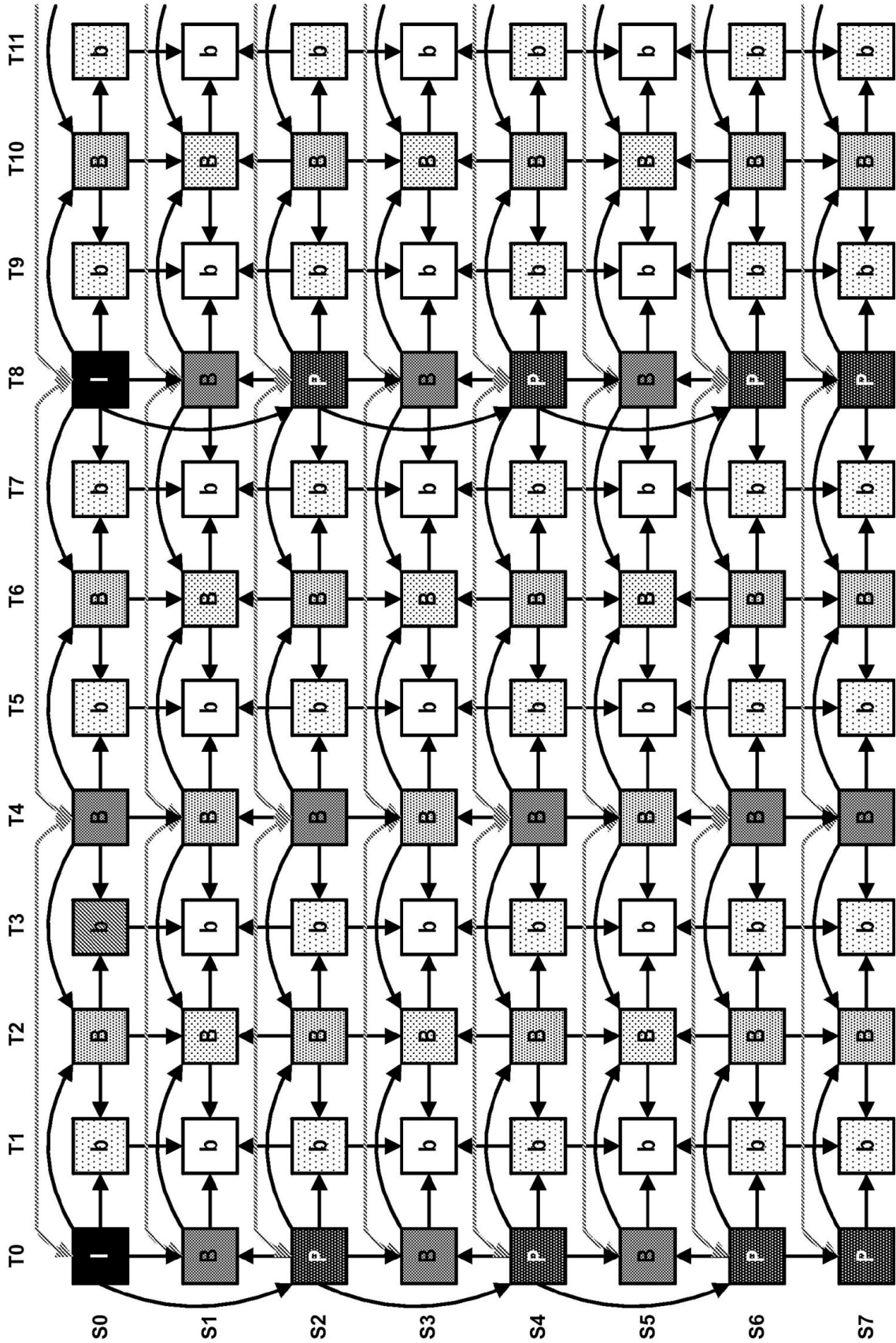


FIG. 3

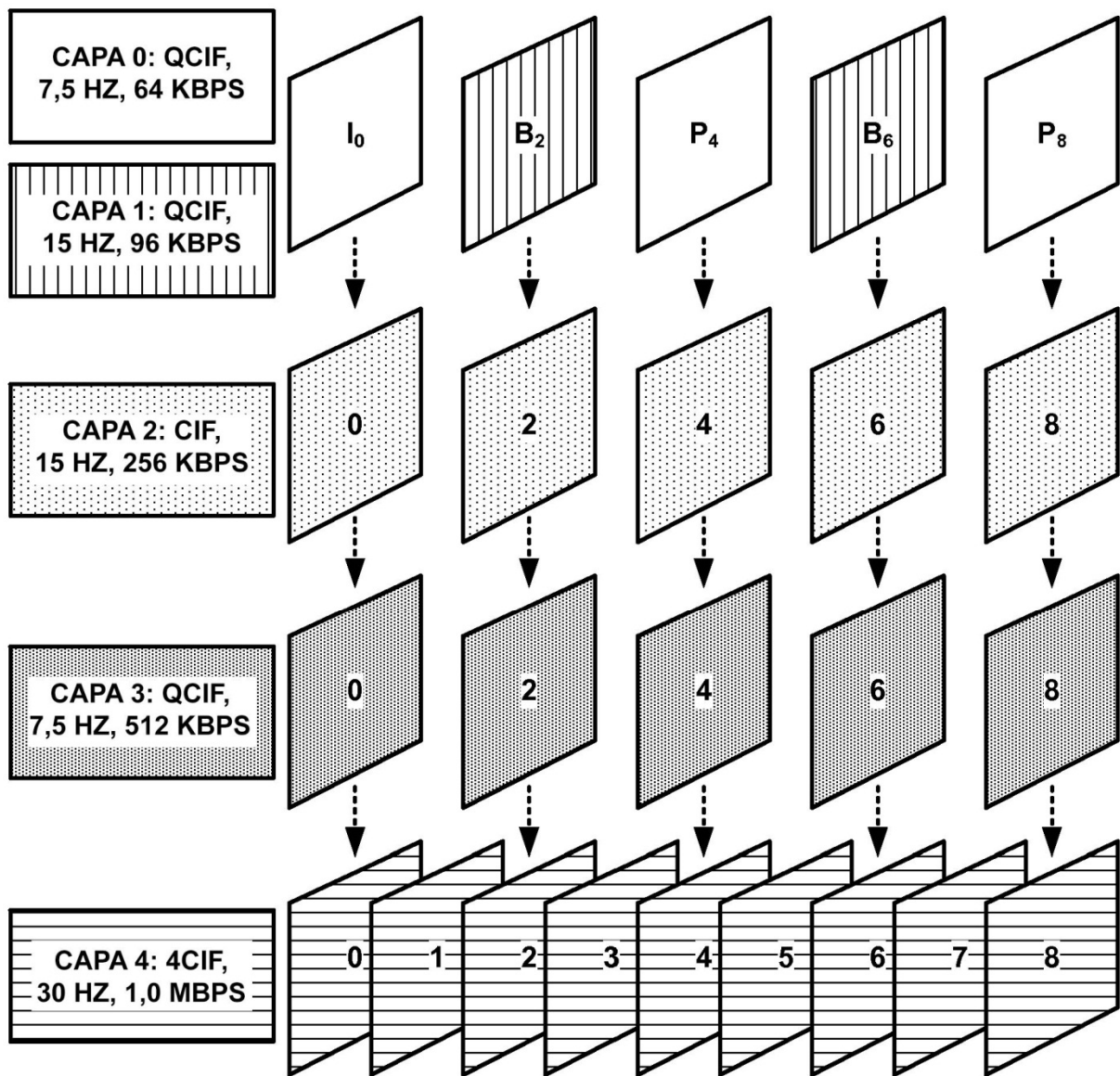


FIG. 4

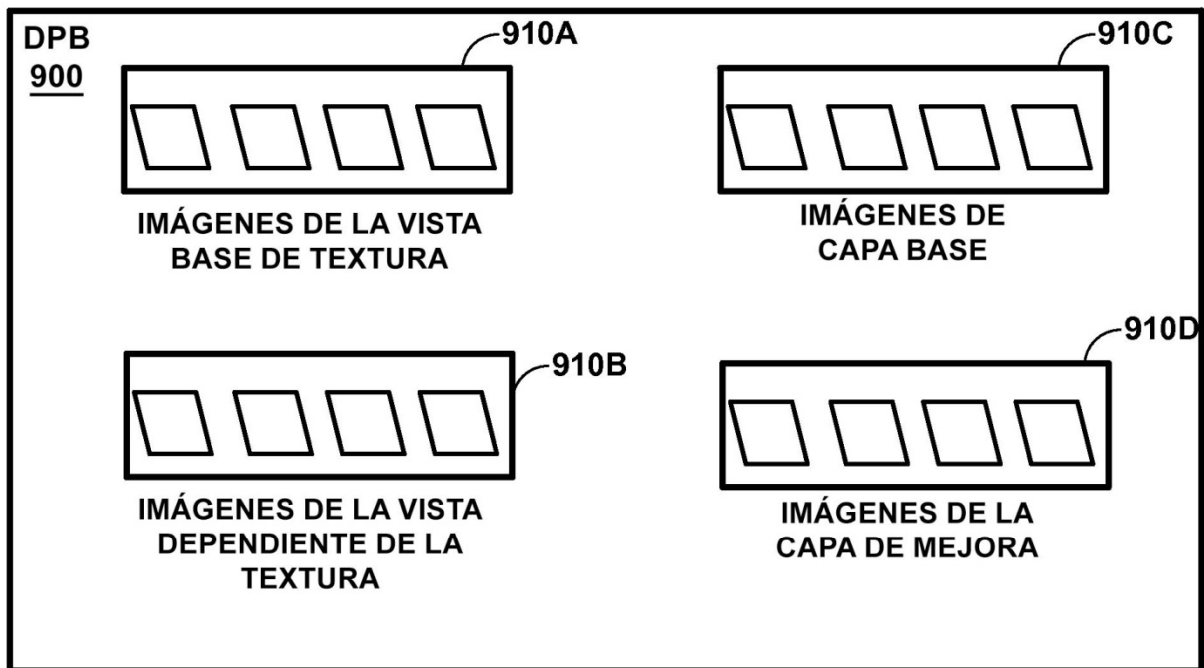


FIG. 5

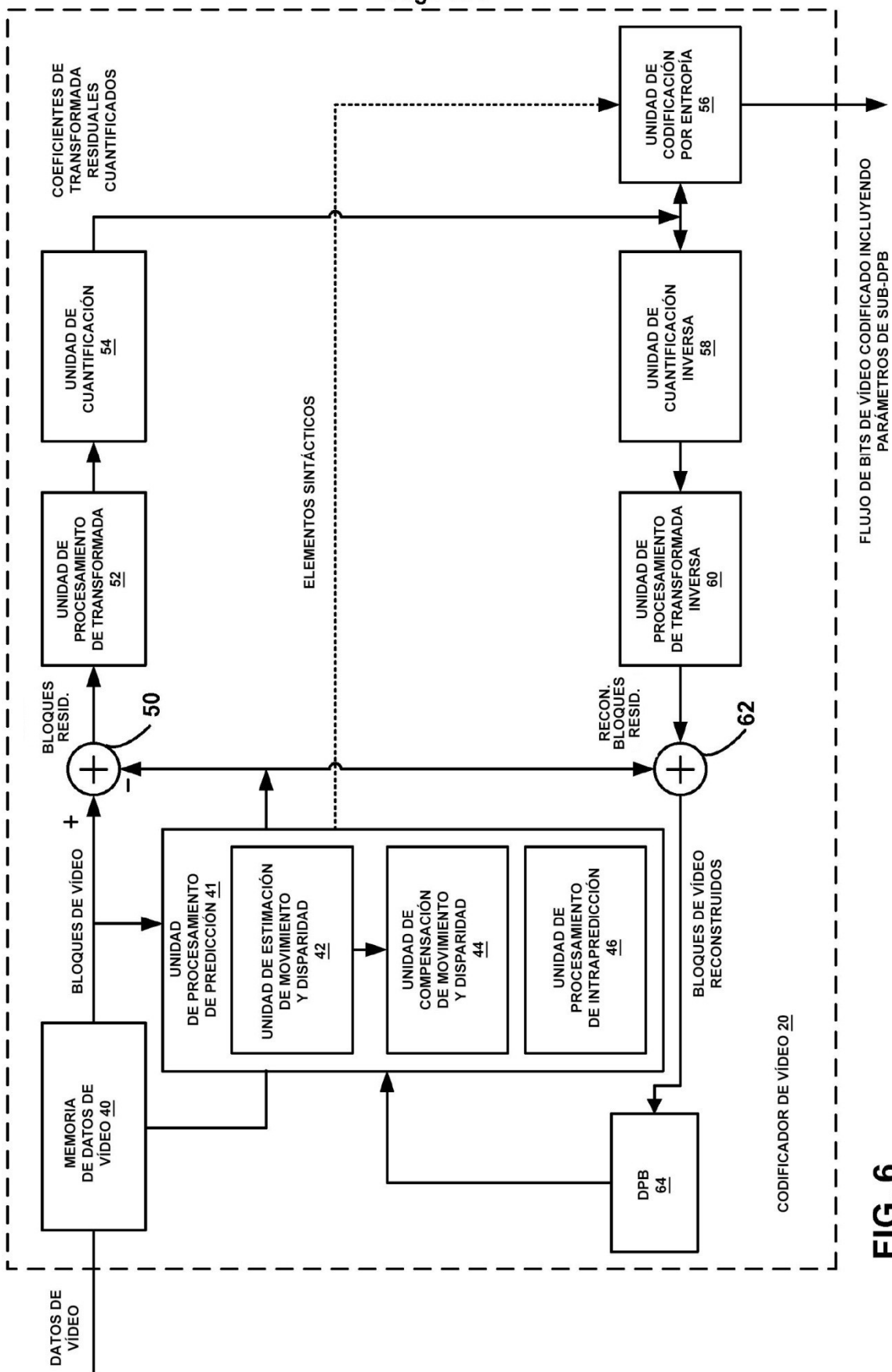


FIG. 6

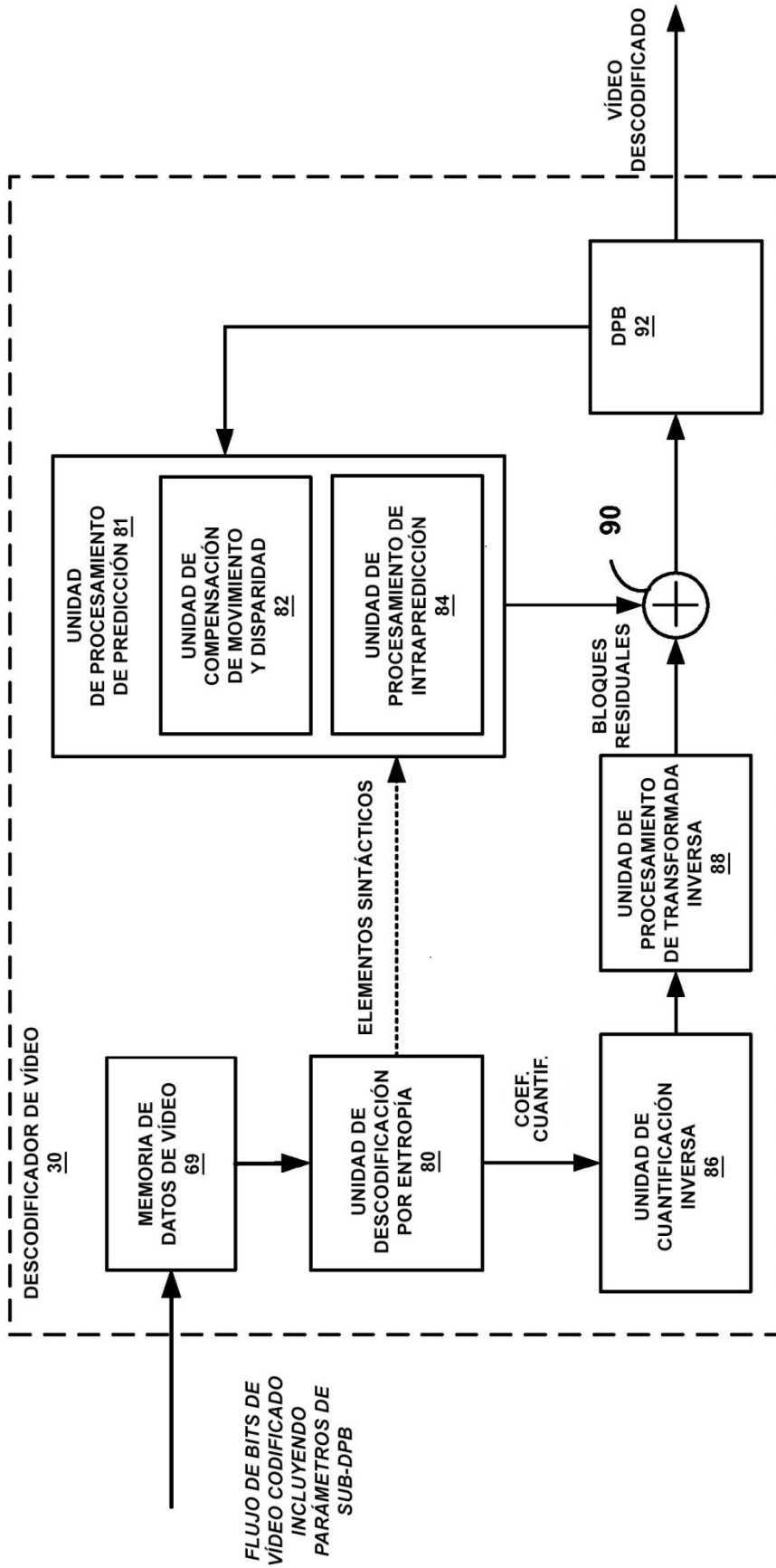


FIG. 7

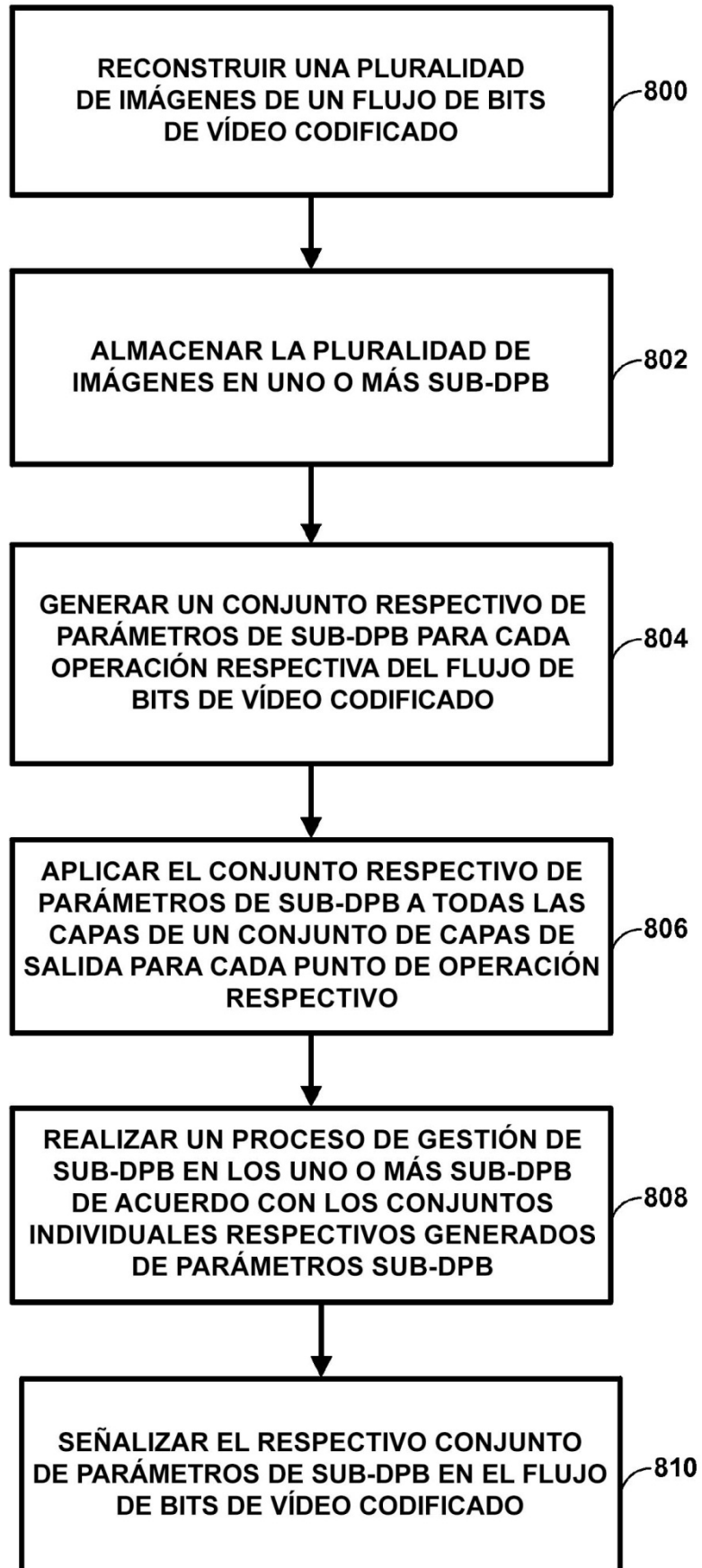


FIG. 8

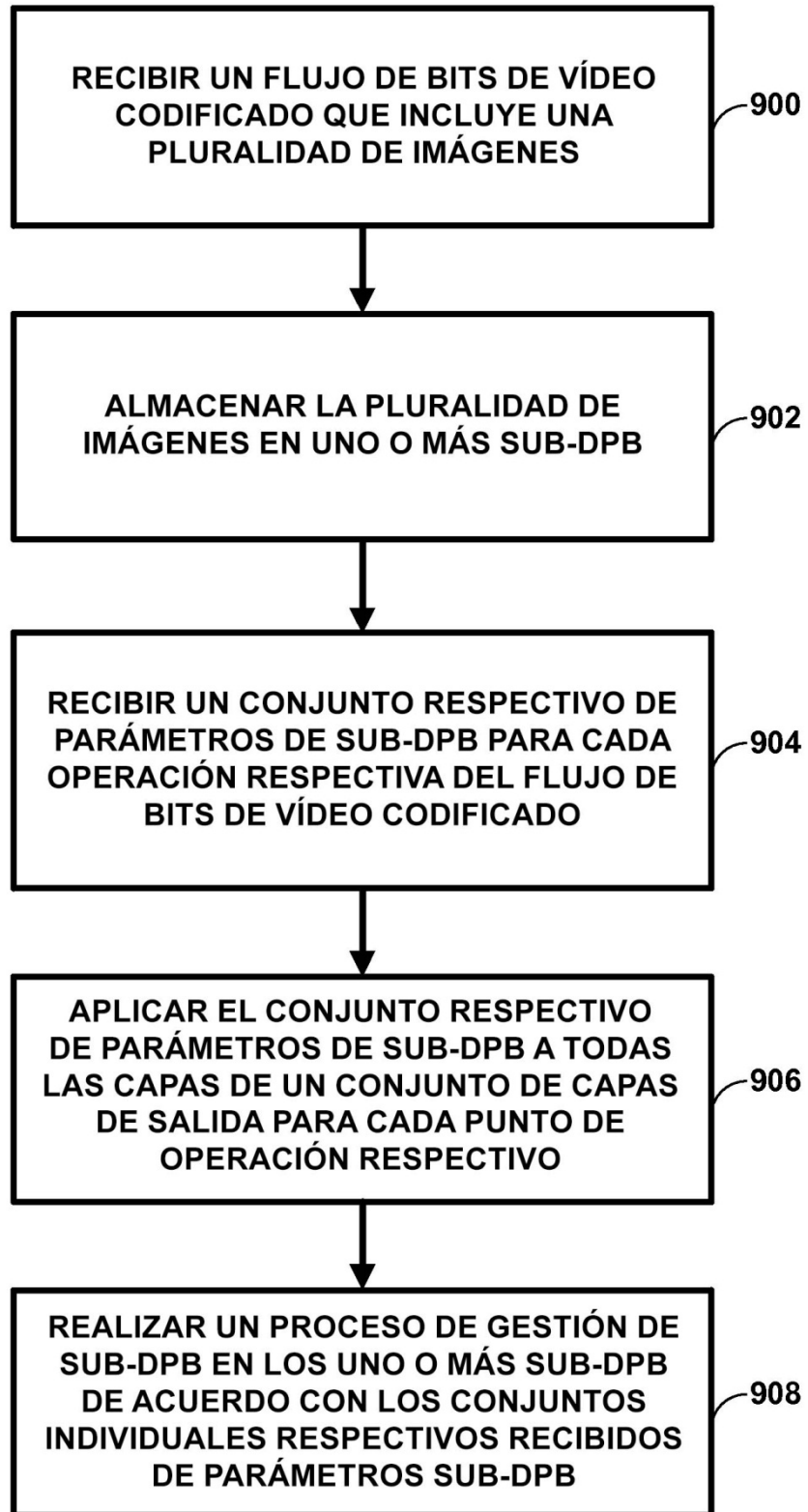


FIG. 9